

Otto-von-Guericke-Universität Magdeburg



Fakultät für Informatik  
Institut für Wissens- und Sprachverarbeitung

## Master Thesis

**Analyse von Texteingaben in einem CAA-Werkzeug zur elektronischen  
Einreichung und Auswertung von Aufgaben**

Verfasser:

Thomas Feustel

10. August 2006

Betreuer:

Prof. Dr. Dietmar Rösner  
Michael Piotrowski, M.A.

Otto-von-Guericke-Universität Magdeburg  
Fakultät für Informatik  
Universitätsplatz 2  
39016 Magdeburg

**Feustel, Thomas:**

Matrikelnummer: 169033

Analyse von Texteingaben in einem CAA-Werkzeug zur elektronischen Einreichung  
und Auswertung von Aufgaben

Master Thesis, Otto-von-Guericke Universität Magdeburg, 2006.

© Thomas Feustel

# Selbstständigkeitserklärung

Hiermit versichere ich, Thomas Feustel (Matrikel-Nr. 169033), die vorliegende Arbeit allein und nur unter Verwendung der angegebenen Quellen angefertigt zu haben.

Thomas Feustel



# Danksagung

Hiermit möchte ich mich bei allen bedanken, die mich während meiner Arbeit an der Master Thesis unterstützten.

Meinen Eltern, Karin und Wieland Feustel, und meinem Bruder Matthias gilt mein besonderer Dank. Sie hatten immer ein offenes Ohr für mich und unterstützten mich auf vielfältige Weise.

Michael Piotrowski, meinem Betreuer an der Otto-von-Guericke-Universität Magdeburg, danke ich für seine intensive Betreuung und den gewinnbringenden Ideenaustausch.

Für ihre aufmunternden Worte, die mich beim Schreiben bestärkten, möchte ich mich bei meinen Freunden Axel, Anne, Markus und Susi bedanken.



# Inhaltsverzeichnis

|           |                                                                      |    |
|-----------|----------------------------------------------------------------------|----|
| Kapitel 1 | Einleitung .....                                                     | 9  |
| 1.1.      | Motivation .....                                                     | 9  |
| 1.2.      | Aufgabenstellung .....                                               | 10 |
| 1.3.      | Gliederung .....                                                     | 11 |
| Kapitel 2 | Einführung in das Gebiet des CAA .....                               | 13 |
| 2.1.      | Gründe für die Benutzung von CAA .....                               | 13 |
| 2.1.1.    | Vorteile von CAA .....                                               | 15 |
| 2.1.2.    | Nachteile von CAA .....                                              | 15 |
| 2.2.      | Objektivität von Tests .....                                         | 16 |
| 2.3.      | Beischreibung einiger Fragetypen .....                               | 16 |
| 2.3.1.    | Auswahlantworttest .....                                             | 16 |
| 2.3.2.    | Mehrfach-Auswahlantworttest .....                                    | 17 |
| 2.3.3.    | Textantwortfragen .....                                              | 17 |
| 2.3.4.    | Zuordnungsaufgabe .....                                              | 18 |
| 2.3.5.    | Richtig/Falsch-Fragen .....                                          | 18 |
| 2.3.6.    | Mathematische Aufgaben .....                                         | 19 |
| 2.3.7.    | Graphical Hotspot .....                                              | 20 |
| 2.3.8.    | Behauptung / Grund .....                                             | 20 |
| 2.3.9.    | Mehrstufige mathematische Aufgaben .....                             | 21 |
| 2.3.10.   | Fragen zu Fallstudien .....                                          | 21 |
| 2.4.      | Zusammenfassung .....                                                | 22 |
| Kapitel 3 | Vorstellung von Softwareprodukten mit CAA-Modulen .....              | 23 |
| 3.1.      | Hotpotatoes .....                                                    | 23 |
| 3.2.      | Blackboard Academic Suite .....                                      | 23 |
| 3.3.      | ILIAS .....                                                          | 24 |
| 3.4.      | EDU Campus / Diploma Campus .....                                    | 25 |
| 3.5.      | Questionmark Perception .....                                        | 25 |
| 3.6.      | Zusammenfassung .....                                                | 26 |
| Kapitel 4 | Konzepte für die Analyse von Texteingaben in ein CAA-System .....    | 27 |
| 4.1.      | Grundlegende Überlegungen .....                                      | 27 |
| 4.1.1.    | Komplexität der Antworten .....                                      | 28 |
| 4.2.      | Freitextantworten .....                                              | 30 |
| 4.2.1.    | Kurzantwort .....                                                    | 30 |
| 4.2.2.    | Freitext in Form von Stichpunktlisten oder Aufsätzen/Essays .....    | 32 |
| 4.2.3.    | Text Mining .....                                                    | 34 |
| 4.2.4.    | Rechtschreibprüfung .....                                            | 35 |
| 4.3.      | Aufgaben aus dem Bereich Mathematik .....                            | 39 |
| 4.3.1.    | Beispiele für mathematische Aufgaben .....                           | 39 |
| 4.3.2.    | Bewertung der eingereichten Lösung .....                             | 40 |
| 4.3.3.    | Eingabe von Formeln bei der Frageerstellung und Lösungseingabe ..... | 42 |

|                      |                                                             |    |
|----------------------|-------------------------------------------------------------|----|
| 4.4.                 | Richtlinien für die Beantwortung von Freitextaufgaben ..... | 44 |
| 4.5.                 | Zusammenfassung .....                                       | 45 |
| Kapitel 5            | Prototypische Implementierung .....                         | 47 |
| 5.1.                 | Programmumgebung .....                                      | 47 |
| 5.2.                 | ECAutoAssessmentBox (ECAAB) .....                           | 48 |
| 5.2.1.               | Voraussetzungen für den Betrieb der ECAAB .....             | 48 |
| 5.2.2.               | Bedienung der ECAAB .....                                   | 49 |
| 5.3.                 | Der Prototyp – TextBackend .....                            | 52 |
| 5.3.1.               | Die Rechtschreibung .....                                   | 53 |
| 5.3.2.               | Die Vorverarbeitung des Textes .....                        | 58 |
| 5.3.3.               | Implementierte Testarten des Prototypen .....               | 62 |
| Kapitel 6            | Zusammenfassung .....                                       | 69 |
| Literaturverzeichnis | .....                                                       | 71 |
| Anhang               | .....                                                       | 77 |
|                      | Blooms Taxonomie von Lernzielen .....                       | 77 |
|                      | Reguläre Ausdrücke .....                                    | 78 |



# Kapitel 1

## Einleitung

Der Titel der Arbeit lautet ‚Analyse von Texteingaben in einem CAA-Werkzeug zur elektronischen Einreichung und Auswertung‘. Das Thema ist im Gebiet des E-Learnings angesiedelt und legt den Fokus auf die von Computern durchgeführte, automatische Auswertung von Aufgaben, die von den Studierenden über speziell eingerichtete Portale abgegeben werden. Die eingereichten Antworten bestehen in diesem besonderen Fall aus Text, der frei formuliert werden kann. Daher wird er auch als Freitext bezeichnet. Der Freitext kann aus einem einzelnen Wort, einer Zeichenketten oder gar einem ganzen Aufsatz bestehen.

### 1.1. Motivation

Während einer Ausbildung fallen bei Leistungskontrollen große Mengen von Ergebnissen an, die einer Überprüfung bedürfen. Für die Lehrenden wäre es deshalb von Vorteil, wenn ihnen ein Tool zur Hand gegeben würde, welches ihnen bei der Kontrolle dieser Ergebnisse zur Seite steht. Bei einer großen Anzahl von Testergebnissen liegt der Schluss nahe, eine computergestützte Auswertung bzw. Bewertung vorzunehmen. Der Forschungszweig CAA (Computer Assisted Assessment) befasst sich mit dieser Art der Datenverarbeitung.

CAA bieten eine Reihe von Vorteilen, die von herkömmlichen Wissensabfragen nicht immer gewährleistet werden können. Sie sind objektiv, konsistent, die Auswertung erfolgt automatisch und sie können problemlos für eine große Anzahl Personen durchgeführt werden. Es ist jedoch schwierig gute Fragen für einen CAA-Test zu erstellen. Aus diesem Grund existieren Richtlinien und Hinweise, die bei der Erstellung von Fragen behilflich sind. CAA-Tests können auch zur Selbsteinschätzung oder Lernförderung von Studenten eingesetzt werden.

Es gibt verschiedene Fragetypen, die mit CAA durchgeführt und ausgewertet werden können. Beispielhaft seien Auswahlantwort- und Zuordnungsfragen genannt (Multiple Choice Questions und Matching-Fragen).

Oft ist es so, dass für die Beantwortung einer Frage eine der folgenden Aktionen durchgeführt werden muss:

- Auswahl der richtigen Antwort aus einer Menge von Antworten
- Identifizieren eines Objektes oder einer Position in einer Grafik (Tabelle, Skizze, Bild, ...)

- Eingabe einer Antwort mittels Tastatur in ein Textfeld

Die Definition der Fragen und Antworten erfolgt in einer festgelegten Art und Weise. Das IMS Global Learning Consortium bietet z.B. mit IMS Question & Test Interoperability (IMS QTI) ein Datenmodell für die Repräsentation von Fragen- und Testdaten an, aus denen wiederum Ergebnisreports erstellt werden können. [IMS]

Die meisten in CAA-Tests benutzten Fragetypen ermöglichen die vollständige automatische Prüfung der eingegebenen Antworten. Allein die Punktevergabe bei falschen/richtigen Antworten zwingt zur Einführung von Bewertungsstrategien. Ein Beispiel wäre eine Auswahlantwortfrage, die aus zwei richtigen und zwei falschen Antwortmöglichkeiten besteht. Wie bewertet man die möglichen Antworten?

Interessant wird es, wenn die Antwort in einem Textfeld eingetragen wird. Die Eingaben können von einfachen Worten/Zahlen bis hin zu komplexen Texten/Gleichungen reichen. Reicht es aus bestimmte Schlüsselworte über reguläre Ausdrücke in diesen Antworten aufzuspüren? Verschiedene Probleme, wie die Benutzung von Synonymen, Beschreibungen oder Rechtschreibfehler und Buchstabendreher, können die Kontrolle weiter erschweren. Eine natürlichsprachliche Untersuchung eines Textes wäre auch denkbar, um weitere Hinweise zur Richtigkeit einer Antwort zu erhalten.

Inwieweit eine in ein Textfeld eingetragene Antwort als 100% richtig eingestuft werden kann, ist schwer abzuschätzen. Bei einfachen Fragen wie 'Wie heißt der Bundespräsident von Deutschland?' ist es noch recht einfach, aber wenn die Aufgabe 'Beschreiben Sie den Prozess der Wundheilung, nachdem man sich mit einem scharfen Gegenstand verletzt hat!' lautet, wird es schon komplizierter. In einem solchen Fall kann das Ergebnis der automatischen Analyse eher der Vorselektion, als der vollautomatischen Kontrolle dienen.

## **1.2. Aufgabenstellung**

Die Master Thesis beschäftigt sich mit der Analyse von Texteingaben in einem CAA-Werkzeug. Zu untersuchen sind die Möglichkeiten solcher Texteingaben und deren Kontrolle. Fragen vom Typ Textantwortfragen und mathematische Ausdrücke bieten sich für die Untersuchung an. Die Idee ist, die Eingaben in das CAA-System in Freitext vorzunehmen und dem Studenten somit größtmögliche Freiheit bei der Beantwortung der Fragen zu geben. Auf der Untersuchung aufbauend soll ein Prototyp implementiert und evaluiert werden. Die Implementierung erfolgt in Python und ist eine Zusatzkomponente der ECAutoAssessmentBox, die unter dem Content Management System Plone zum Einsatz kommt.

### **1.3. Gliederung**

Kapitel 2 der Arbeit bietet einen Überblick über das Gebiet des CAA. Vor- und Nachteile werden benannt und anschließend auf verschiedene Fragetypen eingegangen.

Eine Vorstellung verschiedener Softwareprodukte, die CAA-Module beinhalten, erfolgt in Kapitel 3. Gängige, in Kapitel 2.3 erläuterte, Fragetypen werden von einem Großteil der Beispielsoftware angeboten. Daher wird vorrangig auf die Besonderheiten der einzelnen Produkte eingegangen.

In Kapitel 4 wird auf die speziellen Besonderheiten von Freitextantworten eingegangen. Zu diesem Zweck wird eine Unterscheidung zwischen verschiedenen Typen von Freitextantworten vorgenommen. Für die computergestützte Verarbeitung eines Textes ist seine Freiheit von Rechtschreibfehlern nicht unerheblich. Unterkapitel 4.2.4. befasst sich mit diesem Thema. Weiterhin wird auf die Beantwortung mathematischer Aufgaben gesondert eingegangen und eine Richtlinie für die Beantwortung von Freitextaufgaben vorgenommen.

Kapitel 5 stellt zunächst das webbasiert CAA-Werkzeug ECAutoAssessmentBox vor. Der implementierte Prototyp für die Analyse von Freitextantworten stellt ein Backend für dieses Programm dar. Nachdem Informationen zur Vorverarbeitung einer Textantwort gegeben wurden, wird auf Lösungsmöglichkeiten für die Analyse und Bewertung der Antworten eingegangen.



# Kapitel 2

## Einführung in das Gebiet des CAA

Das Computer-Assisted Assessment wird auch mit anderen Termen, wie. z.B. Computer-Aided Assessment, übersetzt. Der Grundgedanke ist derselbe: die computergestützte Einreichung und Beurteilung von Prüfungsergebnissen.

Im ersten Teil dieses Kapitels wird auf die Gründe eingegangen, die die Benutzung von CAA nahe legen und die Vor- und Nachteile computergestützter Tests beleuchtet. Teil zwei klärt den Begriff der Objektivität im Zusammenhang mit CAA. Anschließend werden im dritten Teil verschiedene Fragetypen vorgestellt.

Ein gutes Beispiel an dem CAA erklärt werden kann, sind Auswahlantwortfragen. Die englische Bezeichnung ist Multiple Choice Questions (MCQ). Auswahlantwortfragen sind Fragen, für die eine Anzahl von Antworten vorgegeben sind. Im einfachsten Fall ist eine Antwort richtig. Diese gilt es herauszufinden. Da von Beginn an klar ist, welche Antwort richtig ist, kann eine automatische Kontrolle der Lösung vorgenommen werden. Weitergehende Untersuchungen über die Lösungen sind ebenfalls möglich.

Wichtig ist die elektronische Erfassung des Datenmaterials. Dies kann entweder direkt durch Eingaben der Antworten am Computer erfolgen (spezielle Software, Webseiten, usw.) oder durch das Einscannen von beantworteten Fragebögen (Papierform). Für das Einscannen werden speziell vorbereitete Vorlagen benutzt, die das Erfassen der Daten mittels OMR (Optical Mark Readers) erleichtern. Die Benutzung von Computern für die Darstellung der Fragen erlaubt den Einsatz von Audio und Video. Dadurch wird eine gute Möglichkeit bereitgestellt, Fragen zu formulieren und Wissen abzufragen.

### 2.1. Gründe für die Benutzung von CAA

CAA bieten eine Reihe von Gründen, die einen Einsatz überlegenswert machen. In [BM04, S.3] werden folgende benannt:

- Die Frequenz der Tests erhöhen
  - Studenten zum Lernen motivieren
  - Studenten ermutigen ihre Fähigkeiten zu trainieren
- Die Bandbreite des zu testenden Wissens erhöhen
- Das Feedback sowohl für die Studenten, als auch die Lehrer erhöhen

- Die Auswahl an Testmethoden erhöhen
- Die Objektivität und Konsistenz erhöhen
- Die Gewichtung der Benotung reduzieren
- Den Verwaltungsaufwand effizienter gestalten

Der Punkt 'Die Gewichtung der Benotung reduzieren' ist dahingehend zu verstehen, dass oft einige wenige Noten zur Benotung herangezogen werden und somit ein starkes Gewicht haben. Wenn CAA mit in den Prozess einbezogen wird, kann durch kontinuierliche Wissensabfragen ein genaueres Bild für die Benotung gezeichnet werden.

CAA kann für formative und summative Evaluierung genutzt werden. Formative Evaluierung erfolgt lernbegleitend, der Student kann CAA nutzen, um seinen Leistungsstand zu überprüfen und entsprechend seine weniger guten Leistungsfelder ausbauen. Die summative Evaluierung dagegen ist vergleichbar mit einer Klausur. Die Leistung des Studenten wird ermittelt und eine Benotung vorgenommen. Einen schönen erklärenden Vergleich von formativer und summativer Evaluierung bietet Robert Stakes mit:

*When the cook tastes the soup, that's formative; when the guests taste the soup, that's summative.*

Weitere tiefgreifende Untersuchungen zu formativer und summativer Evaluierung können bei [Kni02] nachgelesen werden.

CAA ist folglich ein Hilfsmittel, um die Fähigkeiten von Studenten abzu prüfen, aber auch, um sie beim Lernen zu unterstützen. Besonders wichtig ist es die Motivation der Studenten für das Lernen und Üben des Lernstoffes zu erhöhen. Eigene Fortschritte können besser erkannt werden und mit geeignetem Feedback, es sollte akkurat und konstruktiv sein, sind Fehler besser und schneller feststellbar. Das Feedback kann sowohl für die Lehrer als auch für die Studenten hilfreich sein. Der Student erhält ein Feedback in Form von Hinweisen zu seiner Lösung, z.B. Gegenüberstellung der richtigen und falschen Lösungen. Für die Lehrer sind statistische Analysen über die Gesamtantworten mit ihren richtigen und falschen Lösungen interessant. Der Wissenstand der Studenten mit ihren vorhandenen Wissenslücken ist schnell erkennbar und spätere Vorlesungen können daran angepasst werden. Die Art und Weise wie eine Rückmeldung an den Studenten erfolgt, ist vielfältig und hängt von der Art des Tests und dem Fach ab, in dem der Test erfolgte.

Als herausragendste Eigenschaft von CAA kann seine konsistente und objektive Bewertung angesehen werden. Besonders bei großen Gruppen von Studenten kann die automatische Korrektur der Aufgaben sehr hilfreich sein. Man muss sich allerdings vor Augen halten, dass die Erstellung von objektiven Tests kein leichtes Unterfangen ist.

Es folgt eine Zusammenfassung der wichtigsten Vor- und Nachteile nach [BM04, S.3]:

### **2.1.1. Vorteile von CAA**

Durch das automatische Bewerten von Lösungen ermöglicht CAA eine häufigere Durchführung von Tests. Besonders bei größeren Gruppen von Studenten sind Tests mittels CAA einfacher, schneller und konstanter anwendbar. Dies kommt sowohl Lehrern als auch Studenten zu Gute. Lehrer können den Lernfortschritt der Studenten und diese ihre eigenen Fortschritte besser überwachen. Mit den Ergebnissen können diagnostische Reports und Analysen erstellt werden. Da die Tests automatisch bewertet werden, treten keine menschlichen Fehler bei der Korrektur der Arbeiten auf. Eine doppelte Korrektur der Arbeiten ist daher auch unnötig. Detailliertes, spezifisches Feedback kann während und nach dem Test für die Studenten zur Verfügung gestellt werden. Dieses Feedback kann auch Hinweise zur Lösung einer Aufgabe enthalten und die Bewertung entsprechend angepasst werden. Die Möglichkeit Grafiken und Multimedia in die Tests einzubinden, erlaubt das Erstellen von Fragen, die für herkömmliche ‚papierbasierte‘ Tests nur schwer bzw. überhaupt nicht möglich waren. Die Qualität der Fragen kann mit statistischen Untersuchungen einfach überwacht werden. Fragen, die von Studenten oft falsch interpretiert wurden, können abgewandelt werden. Das Einspeisen ‚guter‘ Fragen in Frage-Datenbanken, um sie für spätere Tests wiederholt zu verwenden, bietet sich dabei an.

### **2.1.2. Nachteile von CAA**

CAA-Systeme besitzen allerdings auch Nachteile. Die Anfangsimplementierung eines CAA-Systems kann kosten- und zeitintensiv sein. Die gesamte notwendige Hard- und Software für den Betrieb eines CAA-Systems muss überwacht werden, damit es während der Tests zu keinen Fehlern kommt. Studenten die das CAA-System benutzen sollen, müssen über entsprechende IT-Kenntnisse verfügen, um die Prüfung durchführen zu können. Wenn ein Student erst während eines Tests zum ersten Mal mit einem CAA-System in Kontakt kommt, wird er Zeit für das Erlernen der Bedienung des Systems aufwenden müssen und somit schlechtere Prüfungsergebnisse erzielen. Für die Erstellung der Tests sind in bestimmtem Umfang auch IT-Fähigkeiten notwendig. Bei vielen Fragetypen sind Richtlinien und Tipps einzuhalten, um Fragen zu generieren, die von Studenten nicht missverstanden werden können oder ihnen ein Erraten der richtigen Lösung erleichtern. Die durch CAA unterstützten Fragetypen sind begrenzt. Kapitel 2.3 bietet eine Übersicht.

## **2.2. Objektivität von Tests**

Die mit CAA-Systemen vorgenommenen Tests werden durch das System sachlich und vorurteilsfrei ausgewertet. Das ist möglich, weil die richtigen Antworten vordefiniert sind. Daher spricht man auch von objektiven Tests. Als mögliche Antworten auf solche Tests ist vorstellbar:

- die Auswahl einer oder mehrerer Vorschlagslösungen aus einer Menge von Lösungen
- die Identifizierung eines Objekts oder einer Position

## **2.3. Beischreibung einiger Fragetypen**

Im Folgenden werden verschiedenen Fragetypen kurz vorgestellt. Detaillierte Beschreibungen sind in [BM04, S.29-51] nachzulesen.

### **2.3.1. Auswahlantworttest**

Auswahlantworttests sind meist unter ihrer englischen Bezeichnung 'Multiple Choice Question' (MCQ) bekannt. Bei einer MCQ muss der Student genau eine richtige Antwort aus einer Menge von Antworten auswählen, um die Frage zu beantworten.

Ein Auswahlantworttest wird in vier Elemente unterteilt:

- stem – der Text der Frage
- options – die Antwortmöglichkeiten für die Frage
- key – die richtige Antwort auf die Frage
- distractors – falsche Antwort auf die Frage



**Beispiel [HS91, S.30]:**

Was versteht man unter einem Faradayschen Käfig?

- a) Historischer Blitzableiter
- b) Geschlossene Hülle aus Blech oder Maschendraht, in die von außen kein elektrisches Feld eindringen kann
- c) Physikalische Experimentierbox für Röntgenstrahlen

Die Erstellung der Fragen und Distraktoren ist nicht einfach. Eine Richtlinie mit erklärenden Beispielen kann in [BM04, S.31-36] gefunden werden.

### 2.3.2. Mehrfach-Auswahlantworttest

Mehrfach-Auswahlantworttests oder auch genannt Multiple Response Questions sind eine Abwandlung der MCQ. Die Anzahl der richtigen Antworten ist größer als eins. Das Verschweigen der Anzahl der richtigen Antworten erschwert den Studenten die Beantwortung.

### 2.3.3. Textantwortfragen

Aufgabe bei dieser Art der Frage ist die Beantwortung in Form eines Textes: das Ausfüllen von einem freien Feld mit einem Textstück, einem Wort, einer Zahl oder anderem.

**Beispiele:**

- Der Delphin gehört zur Gattung der \_\_\_.
- Ein Jaguar, ein Gepard und eine Hyäne fressen gemeinsam eine Antilope. Der Jaguar allein würde die Antilope in einer Stunde auffressen. Der Gepard bräuchte drei Stunden dafür und die Hyäne sogar sechs. Wie viel Zeit brauchen sie, wenn sie die Antilope zusammen fressen? Der Jaguar, der Gepard und die Hyäne benötigen \_\_\_ min, um die Antilope aufzufressen. [HS91, S.64]

Bei diesem Fragetyp ist es schwierig die Fragen so zu formulieren, dass nur eine Antwort richtig ist. Probleme bereiten bei der automatischen Kontrolle durch Computer auch Rechtschreibfehler, die Erkennung von Synonymen, Abkürzungen usw. Oft wird nur ein einfacher Stringvergleich zur Feststellung der Korrektheit durchgeführt.

### 2.3.4. Zuordnungsaufgabe

Zuordnungsaufgaben (Matching) bestehen aus einer Reihe von Anweisungen, Statements und Antworten. Aufgabe des Studenten ist es, den Statements ihre Antworten zuzuordnen. Antworten dürfen mehrmals genutzt werden.

**Beispiel:**

Den historischen Ereignissen sollen ihre Jahreszahlen zugeordnet werden.

|                              |      |
|------------------------------|------|
| Beginn erster Weltkrieg      | 9    |
| Beginn zweiter Weltkrieg     | 1914 |
| Schlacht im Teutoburger Wald | 1939 |
|                              | 1918 |

Neben reinen Wissensabfragen eignen sich Matching-Fragen aber auch dafür das Verständnis von Beziehungen und die Anwendung von Wissen zu testen. Im Grunde sind Matching-Fragen eine Variation von MCQ. Wenn bei der Erstellung von MCQ oft die gleichen Antworten verwendet werden, kann man diese Fragen zu Matching-Fragen zusammenfassen.

### 2.3.5. Richtig/Falsch-Fragen

Richtig/Falsch-Fragen sind eine Spezialform von MCQ, bei denen nur zwei Antwortmöglichkeiten zur Verfügung stehen: richtig und falsch. Weitere mögliche Antwortpaare wären auch ja/nein bzw. wahr/falsch. Dieser Fragetyp eignet sich zur schnellen Abfrage großer Wissensbereiche. In Kombination mit MCQ lassen sich komplexere Behauptung/Grund-Fragen (siehe 2.3.8.) erstellen. Problematisch ist die für den Studenten sehr gute Chance die richtige Antwort zu erraten (50%). Um diese Chance zu senken, können mehrere Richtig/Falsch-Fragen zu einer Aufgabe zusammenfasst werden und nur Punkte für eine Aufgabe vergeben, wenn alle enthaltenen Richtig/Falsch-Fragen richtig beantwortet sind. Die Richtig/Falsch-Fragen sollten in diesem Fall das gleiche Thema behandeln. Besonders für komplexe Sachgebiete ist es schwierig Fragen zu stellen, die zweifelsfrei richtig oder falsch sind.

**Beispiel:**

**W F** Die Erde ist eine Kugel.

**W F** Die Erde bewegt sich mit einer elliptischen Bahn um die Sonne.

### 2.3.6. Mathematische Aufgaben

Das Überprüfen mathematischer Antworten ist möglich. Hierbei werden String-Evaluations-Techniken eingesetzt. Oft werden bei diesen Fragen die Variablen für die Aufgabe zufällig erstellt, um Betrugsversuche zu erschweren.

**Beispiel:**

Geben Sie die Formel für die Berechnung des Kreisvolumens an!

### 2.3.7. Graphical Hotspot

Für Graphical Hotspot-Fragen müssen auf einer Grafik Punkte oder Bereiche identifiziert und markiert werden.

**Beispiel:**

Markieren Sie mit dem Fadenkreuz den Oberschenkelknochen (Femur)!  
Bildquelle: [GNetz]



### 2.3.8. Behauptung / Grund

Dieser Fragetyp, auch bekannt als Assertion/Reason, kombiniert Elemente von Auswahlantwortfragen und Richtig/Falsch-Fragen. Auf diese Weise können komplexere Fragen erstellt werden.

Die Frage besteht aus zwei Teilen: einer Behauptung und einem Grund. Zuerst muss der Student überprüfen, ob beide Teile wahrheitsgemäß sind. Ist das der Fall, muss der Student entscheiden, ob der Grund die Behauptung richtig erklärt.

Folgende Konstellationen für den Wahrheitsgehalt von Behauptung und Grund können auftreten:

**Tabelle 2-1 Behauptung/Grund – Varianten der Beantwortung**

| <b>Behauptung</b> | <b>Grund</b> | <b>Auswertung</b>                            |
|-------------------|--------------|----------------------------------------------|
| True              | True         | True - Der Grund ist richtig erklärt.        |
| True              | True         | False - Der Grund ist nicht richtig erklärt. |
| True              | False        | False                                        |
| False             | True         | False                                        |
| False             | False        | False                                        |

Beispiel:

Die Sommer der kommenden Jahre werden immer heißer. *WEIL* Die klimatischen Veränderungen beruhen auf dem den Ausstoß von Kohlendioxid, das bei der Verbrennung fossiler Brennstoffe entsteht.

### 2.3.9. Mehrstufige mathematische Aufgaben

Mehrstufige mathematische Aufgaben (Multi-stage mathematical expression) stellen komplizierte mathematische Fragen dar, die mehr als eine Antwort erwarten. Zusätzlich können auch die Variablen zufällig bei jedem Aufruf geändert werden, so dass jeder Student eine etwas andere Aufgabe lösen muss.

**Beispiel:**  
Kurvendiskussion

### 2.3.10. Fragen zu Fallstudien

Für die Beantwortung der Fragen zu einer Fallstudie (Case Study), muss der Student zuerst eine Fallstudie (ein Szenario, Bericht, Simulation) lesen und verstehen. Anschließend werden Fragen in Form der hier im Kapitel 2.3. vorgestellten Fragetypen gestellt. Für die Beantwortung einer Fallstudienfrage wird daher meist eine sehr viel höhere Zeitspanne benötigt, als bei den bereits vorgestellten Fragetypen.

## 2.4. Zusammenfassung

Das Kapitel bot eine Einführung in den Bereich des CAA und stellte beispielhaft einige Fragetypen vor, die in CAA eingesetzt werden.

In diesem Kapitel wurde nicht darauf eingegangen, welche Lernziele mit den verschiedenen Fragetypen abgefragt werden können. Eine gebräuchliche Einteilung von Lernzielen ist die nach Benjamin Bloom benannte Bloomsche Taxonomie [BM04, S. 23]. Im Anhang ist eine Übersicht der Lernziele nach Bloom zu finden. Eine Untersuchung, inwieweit Lernziele nach Bloom's Taxonomie mit den vorgestellten Fragetypen erreicht werden können, ist in [BM04, S. 22-28] nachzulesen.

In Abschnitt 1.4.1. wurde unter anderem kurz auf die Möglichkeiten des Feedbacks auf eine eingereichte Lösung eingegangen. Einen guten Startpunkt für weiterführende Informationen bietet [BM04, S. 52-62].

Über CAA lässt sich noch vieles mehr berichten, angefangen von der Konstruktion von Test, über die Bewertung und Analyse der Tests, bis hin zu Fragen, die sich mit der Entwicklung und Evaluierung von CAA-Systemen beschäftigen. Für die vorliegende Arbeit war es wichtig einen Eindruck von den Möglichkeiten des CAA zu gewinnen und zu verstehen, dass die verwendeten Fragetypen so gewählt sind, dass ihre Überprüfung automatisch durchgeführt werden kann. Einen Sonderfall bieten Textantwortfragen, da die Möglichkeiten der Eingabe oft sehr begrenzt sind. Meist wird nur ein Wort oder eine Wortgruppe gesucht. Diese müssen zudem genau der Vorgabelösung entsprechen, um als richtig anerkannt zu werden. Wie Textantwortfragen so gestaltet werden können, dass Studenten eine größere Freiheit bei der Beantwortung haben und wie diese Bewertung der eingereichten Antwort vorgenommen werden kann, damit befassen sich die folgenden Kapitel.

# Kapitel 3

## Vorstellung von Softwareprodukten mit CAA-Modulen

Es existiert eine große Anzahl an Softwareprodukten, die unterstützend für die Lehre und Weiterbildung an Schulen, Universitäten, Firmen und anderen Institutionen eingesetzt werden. Ihnen ist gemein, dass sie sehr oft über die Möglichkeit verfügen computergestützte Tests durchzuführen. Dieses Kapitel stellt beispielhaft einige dieser Softwareprodukte vor. Das Hauptaugenmerk liegt auf der Betrachtung von Tests bei denen Texteingaben und deren Auswertung vorgenommen werden.

Für alle Systeme existieren Demo-Versionen, die online bzw. offline ausprobiert werden können.

### 3.1. Hotpotatoes

Hotpotatoes™ ist im Vergleich zu den anderen hier vorgestellten Produkten ein kleines Programm. Sechs Anwendungen stehen zur Verfügung, mit denen interaktiv Auswahlantwort-, Kurzantwort-, Mischmaschsatz-, Kreuzworträtsel-, Zuordnungs- und Lückentextaufgaben für das World Wide Web erstellt werden können.

Für Kurzantwortaufgaben muss eine exakte Übereinstimmung mit einem Wort bzw. einem kurzem Textstück aus einer Vergleichsliste vorliegen, um die Antwort als richtig bewertet zu bekommen. Ebenso ist es bei Lückentexten. Bei Kurzantwortaufgaben wird bei fehlerhafter Eingabe der Antwort, z.B. durch Rechtschreibfehler, ein Hinweis ausgegeben, an welcher Stelle im String der Fehler vorliegt. Dieser kann korrigiert und eine erneute Prüfung durchgeführt werden. Jeder erneute Versuch minimiert jedoch die erhaltende Punktzahl für die Aufgabe. [Hal06]

### 3.2. Blackboard Academic Suite

Die Blackboard Academic Suite™ ist eine Lernumgebung für Netzwerke. Ziel ist es Web-Ressourcen in traditionelle Klassenkurse zu integrieren und alle Kursinformationen für die Studenten jederzeit verfügbar zu halten. Die Software unterteilt sich in ein Learning-, Content- und Community-System, die über ein Web-Interface benutzt werden. Mit dem Learning System führt man Assessments und Umfragen durch. Zur Auswahl stehen alle die in Kapitel 2 vorgestellten Fragetypen.

Am 28. Februar 2005 fusionierten Blackboard Inc. und WebCT, Inc., einem ebenfalls führenden Anbieter für e-learning Systeme, zu einer neuen Firma unter der Marke Blackboard. Sowohl die alten Blackboard, als auch die WebCT Produktlinien blieben erhalten und werden weiter unterstützt. Die besten Features beider Produkte bilden die Grundlage für die neuen Softwareversionen von Blackboard.

Aufsatzfragen und Fragen bei denen eine kurze Antwort als Lösung erwartet wird, werden nicht automatisch korrigiert und müssen vom Lehrer manuell durchgesehen und bewertet werden. Für Lückentexte kann für jedes gesuchte Worte eine Liste möglicher Antworten vorgegeben werden. Eine Rechtschreibprüfung findet nicht statt. Somit muss das Lösungswort korrekt eingegeben werden, um die Aufgabe positiv abzuschließen.

Mathematische Aufgaben werden nach 'Calculated Formula'- und 'Calculated Numeric'-Fragen unterschieden. 'Calculated Numeric'-Fragen erfordern einen numerischen Wert als Antwort. Je nach Einstellung des Aufgabenerstellers muss eine exakte Lösung oder eine Lösung innerhalb eines eingestellten Bereiches vorliegen, um die Aufgabe als richtig bewertet zu bekommen. Die Angabe erfolgt wahlweise numerisch oder in Prozent. Zusätzlich kann auf die gleiche Weise ein weiterer größerer Bereich angegeben werden, der als Teillösung akzeptiert wird. Auf Wunsch kann die Angabe einer Maßeinheit (frei durch den Aufgabenersteller wählbar) definiert und in die Bewertung mit einbezogen werden. 'Calculated Formula'-Fragen bieten mehr Möglichkeiten. Hier gibt der Fragenersteller eine Formel mit Variablen als Musterlösung an. Für die einzelnen Variablen können min- und max-Werte und die Anzahl der Kommastellen definiert werden. Die so erstellten Aufgaben sind nur insofern zufällig, was die Wahl der Variablen und der dazugehörigen Antworten betrifft, dass die Anzahl der möglichen Varianten bei der Fragenerstellung festgelegt wird. Es werden direkt  $n$  zufällig erzeugte Varianten ausgewählt (editierbar) und mit der Aufgabe abgespeichert. [Bla06]

### **3.3. ILIAS**

ILIAS™ ist eine Open Source - Lernplattform und steht für Integriertes Lern-, Informations- und Arbeitssystem. Als CSCL-System (Computer Supported Cooperative Learning) basiert es auf der Verwendung von Computer- und Informationstechnologien sowie dem Internet, um das Lehren und Lernen in Gruppen auch über große Distanzen hinweg zu unterstützen. Die Hauptaufgabe ist die webbasierte Bereitstellung und Erstellung von Lerninhalten.

Eine Auswertung von Freitextantworten ist automatisiert nicht möglich. Lückentexte können computergestützt ausgewertet werden. Dazu müssen alle möglichen Lösungen vorgegeben werden.

Eine Rechtschreibprüfung der eingegebenen Antworten findet nicht statt, aber neben der oft vorhandenen Auswahl, ob die Groß-/Kleinschreibung beachtet werden soll, kann der Wert der Levenshtein-Distanz, mit der die Ähnlichkeit zweier Zeichenketten bestimmt



werden kann, angegeben werden. Die Levenshtein-Distanz gibt die minimale Anzahl an Operationen an, die notwendig sind, um eine Zeichenkette in eine andere überführen zu können. Erlaubte Operationen sind Einfügen, Löschen und Ersetzen. [ILI06]

**Beispiel:**

kaufen    laufen, die Levenshtein-Distanz beträgt 1 (k mit l ersetzt)

### **3.4. EDU Campus / Diploma Campus**

Von der Brownstone Research Group stammen die Produkte EDU Campus™ und Diploma Campus™. Diploma Campus ist eine Erstellungssoftware für Fragen und Tests unter Windows und Macintosh. Diese können in verschiedene Systeme und Formate exportiert werden, z.B. EDU Campus, Blackboard Academic Suite, DHTML. EDU Campus hingegen ist ein Online Testing & Homework Management System und umfasst einen sehr viel größeren Funktionsumfang.

Ebenso wie bei der 'Blackboard Academic Suite' können Aufsatzfragen und Kurzantwortfragen nicht automatisch korrigiert werden. Zusätzlich stehen Kurzphrasenfragen zur Auswahl, die eine noch kürzere Antwort als Kurzantwortfragen erwarten. Diese werden auf exakte Übereinstimmung überprüft. Die große Stärke von EDU Campus im Bereich der Freitextantworten liegt in der Möglichkeit auf mathematische Fragen mit Formeln antworten zu können. Hierzu können die Formeln entweder im Textmodus oder direkt über einen Formeleditor in das Lösungsfeld eingetragen werden. Variablen können bereichs-, listen-, gleichungs-, bildlistbasiert oder konstant sein und für jeden Studenten neu generiert werden. Informationen zu der Anzahl der Kommastellen und zu verwendende Einheiten sind ebenfalls einstellbar. Von allen vorgestellten Produkten sind die Möglichkeiten für mathematische Tests bei EDU Campus und EDU Diploma Campus am beeindruckendsten. [Bro06]

### **3.5. Questionmark Perception**

Questionmark™ Perception™ ist ein Assessment Management System. Im Gegensatz zu den bisher vorgestellten Systemen konzentriert sich Questionmark Perception auf das Verfassen, Planen, Verteilen und Auswerten von Umfragen, Tests, Quizzes und Prüfungen.

Für die Erstellung von Tests stehen 20 verschiedene Fragetypen zur Auswahl. Vier davon benötigen als Antwort eine Texteingabe. Für die automatische Auswertung von Lückentexten, muss eine Liste zulässiger Worte hinterlegt sein. Bei Wortantworten (Antworten, die nur aus einem Wort bestehen) wird ebenfalls auf eine Liste zulässiger

Alternativen zurückgegriffen. Erlaubt sind ein einzelnes Wort, Wortgruppen oder ganze Sätze. Die Bewertung kann abhängig davon gestaltet werden, ob bestimmte Schlüsselworte oder Schlüsselsätze vorhanden sind oder fehlen. Die dafür notwendigen Einstellungen sind kompliziert und die generierte Frage sollte vor der Benutzung ausgiebig getestet werden. Beide Tests sollen auch Rechtschreibfehler erkennen können, was in der Demo-Version des Perception Authorensystems leider nicht nachvollziehbar war. Aufsatzfragen werden auch bei Questionmark Perception normalerweise manuell bewertet. Mit dem externen 'Questionmark Scoring Tool' kann dennoch eine automatische Bewertung mit Hilfe von individuellen Rubriken, Listen zulässiger Antworten, dem Vorhandensein und fehlen bestimmter Schlüsselwörter vorgenommen werden. Für jede Frage können eigene Rubriken und Bewertungsregeln festgelegt werden.

Als Beispielrubriken für einen Geschichtsaufsatz werden Grammatik, Rechtschreibung, Glaubwürdigkeit, Qualität und Recherche genannt [Que06]. Wie das 'Scoring Tool' genau funktioniert, wird aus der Produktbeschreibung nicht klar. Eine telefonische Nachfrage und die Teilnahme an einem Webseminar bei Questionmark konnten die Fragen über die Funktionsweise des 'Scoring Tools' auch nicht klären.

Fragen die eine Zahleneingabe erfordern, lassen nur ganze und reelle Zahlen als Werte zu. Neben der exakten Lösung kann ein Intervall festgelegt werden, der ebenfalls als Lösung für die Aufgabe akzeptiert wird. [Que06]

### **3.6. Zusammenfassung**

Die vorgestellten Systeme lassen für die positive Bewertung einer Texteingabe meist nur eine exakte Übereinstimmung mit einer Vorgabeliste von Wörtern zu. Kleinere Rechtschreib- oder Tippfehler (Buchstabendreher, Buchstabe zuviel/zuwenig) werden von ILIAS durch die Levenshtein-Distanz erkannt. Questionmark Perception benutzt laut Dokumentation anscheinend ein ähnliches Verfahren. Bis auf Questionmark bietet keines der Systeme die automatische Bewertung von Aufsatzfragen an. Die Bewertung wird allerdings durch ein externes Tool durchgeführt und konnte nicht getestet werden.

Mathematische Aufgaben werden bei einem Teil der Systeme als spezielle Fragetypen angesehen und bieten darauf angepasste Bewertungsoptionen, wie z.B. die Angabe eines Bereiches, in der die Lösung liegen darf. Die von EDU Campus bereitgestellten Möglichkeiten zur Eingabe von Formeln als Antwort auf eine Frage sind sehr gut und wird von keinem anderem der vorgestellten Softwareprodukte angeboten.

# Kapitel 4

## Konzepte für die Analyse von Texteingaben in ein CAA-System

### 4.1. Grundlegende Überlegungen

CAA-Systeme nutzen für die Durchführung von Tests Fragetypen, bei denen die richtige Lösung im Zuge der Aufgabenerstellung mit angegeben wird. Wird ein Test eingereicht, vergleicht das System die Lösungen des Studenten mit den vorgegebenen Lösungen und kann automatisch eine Bewertung vornehmen. Auswahlantwortfragen sind dafür ein sehr gutes Beispiel. Gegeben sind Fragen mit ihren Antwortmöglichkeiten. Nur eine der Antworten ist korrekt und soll durch die Studenten ausgewählt werden.

**Beispiel:**

Was versteht man unter Ornithologie?

- a) Insektenkunde
- b) Paarungsverhalten
- c) Vogelkunde

Während der Aufgabenerstellung wurde im CAA-System die richtige Antwort hinterlegt. Wie die richtige Antwort hinterlegt wird, hängt von der Implementierung des CAA-Systems ab. Die Tutorial Markup Language (TML) oder IMS Question & Test Interoperability (IMS QTI) hinterlegen die Informationen zu einem Test in Form von XML-Daten. Das CAA-System führt nun einen Vergleich der studentischen Lösung mit der Vorgabelösung durch und kann automatisch den Test kontrollieren. [IMS], [TML]

Komplizierter wird es bei den in 2.3.3 vorgestellten Textantwortfragen. Für ihre Beantwortung muss der Student eine einfache Antwort auf eine Frage geben oder einen Lückentext mit Wörtern oder Zahlen füllen.

**Beispiele:**

- a) Alle \_\_ Jahre ist ein Schaltjahr.
- b) Welches Land trat 1941 in den zweiten Weltkrieg ein, nachdem Japan Pearl Harbor überfiel?

Als Vergleichslösung werden für a) ‚4‘ und für b) ‚USA‘ hinterlegt. Für a) ist die Prüfung auf Korrektheit einer eingereichten Lösung noch einfach. Anders sieht es bei Aufgabe b) aus. Tippfehler, Rechtschreibfehler oder die Verwendung von Synonymen kann eine eingereichte Lösung für das CAA-System falsch erscheinen lassen. Geht man einen Schritt weiter und lässt den Studenten als Lösung für eine Aufgabe einen Text eingeben, kommen zusätzliche Probleme hinzu. Die Antworten auf eine Frage können sich stark unterscheiden, sowohl im Umfang als auch an Komplexität. Es gibt keine klare Trennung zwischen wahr und falsch, vielmehr ist die Antwort zu einem gewissen Grad richtig. Die Spanne reicht von 0-100%.

Die Analyse von Texten ist ein Teilgebiet der Computerlinguistik. Sie bildet die Schnittstelle zwischen Sprachwissenschaft und Informatik. Ziel ist es natürliche Sprache (gesprochene wie geschriebene) mit Hilfe von Computern Verarbeiten und Auswerten zu können. In den folgenden Kapiteln wird auf einige spezielle Verfahrensweisen der Computerlinguistik eingegangen und ihre Nützlichkeit in Hinsicht auf die Auswertung in ein CAA-System eingereichten Antworten eingegangen.

#### **4.1.1. Komplexität der Antworten**

Mit steigender Komplexität der Antworten wird es immer schwieriger eine programmgesteuerte automatische Entscheidung hinsichtlich ihrer Richtigkeit zu treffen. Die Bewertung der Antworten entspricht den allgemein bekannten Maßstäben. Im schulischen Bereich wird oft eine Skala von 1 (sehr gut) bis 6 (ungenügend) verwendet. Das entspricht einer gröberen Einteilung der prozentualen Richtigkeit.

Antworten die aus einem einzelnen Wort bestehen, können über einen direkten Stringvergleich überprüft werden. In dem obigen Beispiel wird die Zeichenkette ‚USA‘ als Antwort auf die Frage erwartet. Um Fehler, die Aufgrund der Groß-/Kleinschreibung auftreten zu vermeiden, könnte die interne Verarbeitung alle Zeichenketten in Kleinbuchstaben umwandeln. Trotzdem kann es immer noch passieren, dass als Lösung ein Synonym von ‚USA‘ eingegeben wird, z.B. ‚United States of America‘. Über einen Abgleich mit einer Art Synonymwörterbuch ist auch dieses Problem recht schnell behoben. Aber selbst jetzt können noch Probleme auftauchen, wenn die Antwort Rechtschreibfehler enthält: ‚Unitet States of America‘. ‚Unitet‘ ist fälschlich mit einem ‚t‘ am Ende geschrieben und führt dazu, dass die

Antwort als falsch bewertet wird. Über eine Rechtschreibprüfung kann auch dieses Problem gelöst werden. Es stellt sich allerdings die Frage, wie sie durchgeführt wird. Am Einfachsten und wahrscheinlich auch am Besten wäre eine interaktive Rechtschreibprüfung, bei der der Student die Fehler korrigiert. Eine automatische Rechtschreibprüfung durch das CAA-System wird Rechtschreibfehler in einem gewissen Maße erkennen, ob deren Korrektur aber die gewünschten Ergebnisse liefert, muss geprüft werden. In Kapitel 5.2.3. findet sich eine Evaluierung von PyEnchant, einer Programmbibliothek für die Rechtschreibprüfung unter Python.

Wird für die Beantwortung der Frage mehr als ein Wort benötigt, reicht ein bloßer Stringvergleich mit einer Vergleichslösung nicht aus. Zu verschieden können die Antworten ausfallen und dennoch richtig sein. Wie sieht in diesem Fall eine Vergleichslösung aus? Aus Sicht des Fragenerstellers ist es wünschenswert, so wenig wie möglich Zeit mit der Erstellung einer Frage zu verbringen, also auch in möglichst kurzer Zeit eine Musterlösung zu erzeugen. Im Idealfall kann die Musterlösung durch einfaches ‚Copy & Paste‘ der entsprechenden Texte der Unterrichtsmaterialien erstellt werden.

Für Fragen die mathematische Antworten erwarten, muss sichergestellt werden, dass das Runden von Zwischenergebnissen und damit ein leicht abweichendes Endergebnis mit abgefangen wird. Die Eingabe von Formeln oder anderen mathematischen Ausdrücken, die eine Lösung auf die Frage darstellen, ist ebenfalls wünschenswert.

In jedem Fall ist es ratsam sowohl die eingereichte Lösung, als auch die Musterlösung in eine Repräsentation zu überführen, die einen Vergleich ermöglicht. Rechtschreibfehler sollten dabei weitgehend korrigiert, unnütze Wörter entfernt und jedes Wort auf entsprechende Synonyme überprüft sein. Je mehr Informationen in der Antwort integriert und von der Musterlösung gefordert werden, desto schwerer ist eine Beurteilung in Richtung richtig und falsch. Die eingereichte Antwort wird die Frage in einem gewissen Umfang beantworten und die Aufgabe des Analyseprogramms ist es diesen zu ‚berechnen‘. Das Ergebnis kann ein Wert zwischen 0 (falsch) und 1 (richtig) sein, z.B. der Anteil benutzter Schlüsselworte einer in der Musterlösung hinterlegten Stichwortliste, die für das zu behandelnde Thema charakteristisch sind. Wenn die Antwort in Form eines Essays oder Aufsatzes abgegeben wird und eine Meinung des Studenten enthält, kann keine klare Aussage über ihre Richtigkeit getätigt werden, da die Meinungen der Menschen stark variieren und nicht alle in einer Musterlösung hinterlegt werden können. Zusätzlich kann eine Auswertung der Rechtschreib- und Grammatikfehler, Anzahl verwendeter Wörter/Sätze, die durchschnittliche Länge der Wörter/Sätze oder des Schreibstils vorgenommen werden. Demnach ist das Ergebnis der Analyse nicht nur eine Aussage über die Richtigkeit einer Lösung (sofern sie berechnet werden konnte), sondern auch eine Hilfestellung für den Lehrenden, der die eingereichten Antworten korrigieren oder benoten muss. Bei einer Vielzahl von Antworten, die nur stickprobenartig durchgesehen werden brauchen, könnte er z.B. mit Hilfe der Analyseergebnisse Einreichungen identifizieren, die seine besondere Beachtung finden sollten.

## 4.2. Freitextantworten

Freitextantworten werden in dieser Arbeit in zwei Kategorien unterteilt: Kurzantwort und Freitext in Form von Stichpunktlisten oder Aufsätzen/Essays. Kurzantworten zeichnen sich durch ihre Kürze, oft nur ein Wort, Name oder Wortgruppe, aus und sind relativ einfach automatisch zu überprüfen. Direkte Freitextantworten bestehen dagegen aus beliebig langen Texten (Stichpunkte, Aufsätze) und stellen viel höhere Anforderungen an ihre Verarbeitung. Mit zunehmender Komplexität der Antworten nehmen die Probleme der Verarbeitung zu und erfordern weiterreichende Verarbeitungsschritte, um eine Aussage über die Korrektheit der Lösung zu erhalten.

Im Folgenden werden verschiedene Beispielaufgaben mit ihren Musterlösungen und beispielhaft eingereichten Lösungen gezeigt. Nach jedem Beispiel wird auf mögliche Probleme bei der automatischen Verarbeitung hingewiesen und Möglichkeiten der Definition einer geeigneten Musterlösung diskutiert.

### 4.2.1. Kurzantwort

#### Einzelnes Wort

**Beispiel:**

Frage: Wo kam es erstmals zur Verwendung des Begriffes Imperialismus?

Musterlösung: Frankreich

Eingereichte Lösung: Frankreich

Das ist die einfachste Möglichkeit einer Texteingabe. Durch einen direkten Stringvergleich kann die Korrektheit der Lösung festgestellt werden. Rechtschreibfehler und Synonyme sind einfach zu überprüfen. Um eine genaue Antwort zu erzielen, muss die Frage sehr präzise gestellt werden und möglichst wenig Raum für andere Antwortmöglichkeiten lassen. Lückentexte sind eine andere weitaus effektivere Variante einzelne Worte abzufragen, da der sie umgebende Text bei der Wahl des einzutragenden Wortes keine so weiten Möglichkeiten bietet.

Als Musterlösung reicht das gesuchte Wort aus. Mit seiner Hilfe können bei Bedarf die Synonyme erzeugt werden, die mit der eingereichten Lösung verglichen werden, wenn die eingereichte Lösung den direkten Vergleich mit der Musterlösung nicht besteht.

## Wortgruppe

**Beispiel:**

Frage: Wann kam es erstmals zur Verwendung des Begriffes Imperialismus?

Musterlösung: Mitte des 19. Jahrhundert

Eingereichte Lösung: Mitte des 19. Jh.

Die Möglichkeiten mit vermehrten Problemen bei der Überprüfung einer Lösung konfrontiert zu werden, steigt bei Wortgruppen gegenüber einzelnen Worten enorm an. Sobald mehr als ein Wort als Antwort erwartet wird, muss mit der Verwendung von Abkürzungen gerechnet werden. Solange diese konform zu Abkürzungen in einem Wörterbuch sind, spielt das keine Rolle. Sind die Abkürzungen aber das Resultat sehr schnellen Schreibens, kommt es häufig vor, dass Abkürzungen auftreten, die sehr willkürlich sind bzw. allgemein genutzt werden, aber nicht im Wörterbuch enthalten sind. Zusätzlich gibt es Abkürzungen mit mehreren Bedeutungen, die sich oft nur aus dem Sinnzusammenhang erschließen.

**Tabelle 4-1 verwendete Abkürzungen in Freitexten**

| <b>Beispielwort</b> | <b>Abkürzung</b> | <b>enthalten im [Duden96]</b> | <b>nicht enthalten im [Duden96]</b> |
|---------------------|------------------|-------------------------------|-------------------------------------|
| beziehungsweise     | bzw.             | x                             |                                     |
| Jahrhundert         | Jh.              | x                             |                                     |
| wissenschaftlich    | wissenschaftl.   |                               | x                                   |
| Großmächte          | GM               |                               | x                                   |
| Volkshochschule     | VHS              | x                             |                                     |
| Video Home System   | VHS              |                               | x                                   |

Im Tabelle 4-1 wurde Jahrhundert mit Jh. abgekürzt. Mittels eines Wörterbuches kann die Abkürzung durch das unabgekürzte Wort ersetzt und besser mit der Musterlösung verglichen werden.

## 4.2.2. Freitext in Form von Stichpunktlisten oder Aufsätzen/Essays

### Stichpunkte

#### **Beispiel:**

Frage: Stellen Sie alte und neue Großmächte des Imperialismus gegenüber und geben Sie Ursachen dafür an, dass Großbritannien den ersten Platz inne hatte!

Musterlösung: Die alten Großmächte (Spanien, Frankreich, Großbritannien und Portugal) bekamen durch die neuen Großmächten (Deutschland, Russland, USA und Japan) zusätzliche Konkurrenz. Betrachtet man alle Großmächte, nimmt Großbritannien eine Sonderstellung zwischen ihnen ein. Seine technische und wirtschaftliche Entwicklung ist am weitesten fortgeschritten und die nationalstaatliche Bildung war frühzeitig abgeschlossen. Das frühe Errichten von Kolonien versorgte Großbritannien zusätzlich mit den größten und lukrativsten kolonialisierten Gebieten.

#### Eingereichte Lösung:

alte GM:

- Spanien
- Frankreich
- *Großbritannien*
- Portugal

neue GM:

- Deutschland
- Russland
- USA
- Japan

GB 1. Platz:

- technische und *wirtschaftl.* am fortgeschrittensten  
Entwicklung der Dampfmaschine
- *nationalstaatl.* Bildung frühzeitig abgeschlossen
- begann schön früh (früher als neue Großmächte) Kolonien zu errichten

Wird der Antworttext länger, tauchen vermehrt Rechtschreibfehler auf (Großbritannien). Des Weiteren werden Abkürzungen verwendet, die nicht zwingend in Wörterbüchern hinterlegt sind (z.B. GM für Großmächte).

Stichpunkte lassen sich sehr gut von einander unterscheiden und damit die einzelnen Aussagen von einander trennen. Die Erkennung von Unterstichpunkten kann durch



Einrückungen im Text und Verwendung unterschiedlicher Stichpunktzeichen durchgeführt werden.

## Aufsatz

### **Beispiel:**

Frage: Definieren Sie den Begriff Imperialismus im engeren Sinne!

Musterlösung: Der Imperialismus bezeichnet die Expansionspolitik europäischer Großmächte in Afrika und Asien im letzten Drittel des 19. Jahrhunderts und dem ersten Weltkrieg.

Eingereichte Lösung: Bezeichnet Expansionspolitik *europäischer* Staaten (GB, D, F, Russland sowie Japan, USA) in Afrika und Asien im letzten Drittel des *19Jh* bis zum 1. Weltkrieg.

Aufsätze stellen die komplexeste Version einer eingereichten Lösung dar. Ihre Kontrolle hinsichtlich einer Aussage über Richtig und Falsch ist nur bedingt möglich. Während in Stichpunktlisten jeder Stichpunkt eine eigene Aussage definiert, sind die Aussagen innerhalb eines Aufsatzes nicht so leicht zu unterscheiden. Die Unterteilung des Textes in Absätze und später in Sätze erlaubt es eine Strukturierung der Aussagen vorzunehmen. Natürlich ist es auch vom Autor eines Aufsatzes abhängig, wie gut sein Text gegliedert ist.

Abgesehen von den inhaltlichen Aussagen eines Textes, können weitere Informationen über den Text gesammelt werden, die mehr statistischer Natur sind:

- Länge des Textes
- durchschnittliche Wort- und Satzlänge
- Anteil der Rechtschreibfehler
- Anzahl der Kommas / Kommas pro Satz
- Verwendung stilistischer Mittel

Der klare Vorteil statistischer Aussagen ist ihre Vergleichbarkeit. Entsprechend vorbereitete Statistiken erlauben das schnelle Erkennen und Einschätzen von eingereichten Lösungen. Soll zum Beispiel eine Hausarbeit mit einem Umfang von ca. fünf Seiten abgegeben werden, kann an Hand der benutzten Wörter in der Arbeit abgeschätzt werden, in wie weit fünf Seiten Text geschrieben wurden.

### 4.2.3. Text Mining

Für die Analyse und Auswertung der eingereichten Lösung können Bearbeitungstechniken aus dem Bereich Text Mining eingesetzt werden. Ziel des Text Minings ist es mit Hilfe von Computern automatisiert neue und relevante Informationen aus Texten zu gewinnen. Dazu werden statistische und linguistische Methoden angewendet. Grundsätzlich werden zwei Techniken des Text Minings unterschieden [UniBe]:

- dokumentenbasierte Techniken
- inhaltsbasierte Techniken

Dokumentenbasierte Techniken arbeiten mit Dokumentensammlungen unterschiedlichen Umfangs. Beispiele für diese Techniken sind:

- Textkategorisierung  
Es wird zwischen der automatischen Kategorisierung, bei der die Kategorien vorgegeben sind und die Texte ihnen zugeordnet werden, und dem Clustering unterschieden. Beim Clustering sind die Kategorien nicht vorgegeben und werden durch das Zusammenfassen von Dokumenten gleichen Inhalts erzeugt.
- Information Retrieval  
Das Information Retrieval dient dazu diejenigen Dokumente aus einer Dokumentensammlung auszusuchen, die am Besten der Anfrage eines Benutzers entsprechen. Internetsuchmaschinen (z.B. Google) verwenden Methoden des Information Retrieval.

Inhaltsbasierte Techniken des Text Minings sind dagegen nur auf ein Dokument ausgerichtet und nicht auf eine ganze Dokumentensammlung. Textzusammenfassungen sind ein Beispiel dafür. Sie fassen den Text zusammen und ermöglichen es schnell einen Überblick über seinen Inhalt zu erhalten und zu entscheiden, ob es sich lohnt ihn komplett durchzulesen.

Für die Auswertung der eingereichten Lösungen sind dokumentenbasierte Techniken auf den ersten Blick ungeeignet, weil sie eine Dokumentensammlung benötigen auf die sie für die Analyse zugreifen. Als Vergleichsdokument steht nur die Musterlösung zur Verfügung, die dann z.B. die einzige Kategorie darstellt. Die Ähnlichkeit der eingereichten Lösung zur Musterlösung wäre dann der Grad ihrer Richtigkeit. Es stellen sich in dieser Konstellation jedoch schnell Probleme ein. Beispielsweise ist die bekannte Gewichtungsmethode Tf-Idf-Maß (term frequency – inverse document frequency) nicht einsetzbar. Tf gibt die Häufigkeit eines Terms innerhalb eines Dokumentes an, ist jedoch nur eine ungenügende Information darüber, wie charakteristisch der Term für ein Dokument ist. Hier kommt die Idf ins Spiel. Sie gibt

Auskunft darüber, in wievielen Dokumenten der Dokumentenkollektion ein Term enthalten ist. Ist ein Term in einem Dokument sehr häufig vorhanden, aber nur in wenigen Dokumenten der Dokumentenkollektion, so ist er sehr charakteristisch für dieses Dokument. Die Berechnungsformel für das Tf-Idf-Maß lautet [Eul01]:

$$w_{ik} = tf_{ik} * \log_2\left(\frac{N}{n_k}\right)$$

$w_{ik}$  ist das Tf-Idf-Gewicht des  $k$ -ten Termes im  $i$ -ten Dokument. Dabei sei  $N$  die Anzahl der Dokumente in der Dokumentenkollektion,  $n_k$  die Anzahl der Texte, in denen der Term  $t$ , vorkommt und  $tf_{ik}$  die Häufigkeit des Terms im Dokument  $i$ .

Berechnet man das Tf-Idf-Gewicht aller Terme der eingereichten Lösung und nimmt als Dokumentenkollektion nur die Musterlösung heran, so ergibt sich immer das gleiche Gewicht von 0. Dieses Ergebnis der Gewichtung ist für eine Analyse nicht zu gebrauchen.

Die Zusammenfassung einer eingereichten Lösung mittels spezieller Methoden inhaltsbasierter Techniken ist nur bedingt nützlich, da sie für die Auswertung der eingereichten Lösung nicht notwendig ist. Werden die aus der Zusammenfassung erstellten Texte für die Prüfung auf Richtigkeit der eingereichten Lösung genutzt, besteht die Gefahr, dass wichtige Teile der Lösung gekürzt werden und somit die Bewertung mindern. Sind eingereichte Aufsätze sehr lang, ist die automatische Erstellung einer Zusammenfassung für den Lehrenden interessant, um sich eine schnelle Übersicht über mehrere eingereichte Aufsätze zu verschaffen.

#### 4.2.4. Rechtschreibprüfung

Sobald ein Feitext niedergeschrieben wird, besteht die Gefahr, dass sich Rechtschreibfehler einschleichen. Für zu analysierende Texteingaben wird davon ausgegangen, dass die Texte orthographisch einwandfrei sind, da es die Verarbeitung erheblich vereinfacht. Dies gilt sowohl für die Texteingaben des Studenten, als auch für die Texteingaben des Aufgabenerstellers im Bereich der vordefinierten Lösung. Es ist deshalb sehr hilfreich, vor der Verarbeitung eines Textes eine Rechtschreibprüfung durchzuführen.

Die Rechtschreibprüfung gliedert sich in zwei Teile: die Fehlererkennung und die Fehlerkorrektur. Im Rahmen der Fehlererkennung wird untersucht, ob eine Zeichenkette in einer vorgegebenen Wortliste, einem Lexikon oder einem Wörterbuch enthalten ist. Die Fehlerkorrektur gestaltet sich schwieriger, da für ein fehlerhaftes Wort oft eine Reihe von Korrekturkandidaten erzeugt wird, welche nach ihrer 'Richtigkeit' geordnet sind. Die Reihenfolge der Kandidaten ist von dem verwendeten Sortierungssystem abhängig. Hinzu kommt, dass ein Wörterbuch einem stetigen Wandel unterliegt und

ständig aktualisiert werden muss. Die Korrektur kann interaktiv durch den Nutzer erfolgen, der aus der vorgeschlagenen Liste von Kandidaten das richtige Wort wählt oder die Korrektur erfolgt automatisch. Im letzteren Fall stellt sich die Frage, welches Wort aus der Vorschlagsliste für die Korrektur herangezogen wird. Nicht in jedem Fall ist es das erste Wort in der Liste. In [Kuk92] wird für die ersten drei Vorschläge eine Treffgenauigkeit von über 90% genannt.

In [Kuk92] werden drei Probleme bei der automatischen Korrektur von Wörtern besonders hervorgehoben:

- Erkennung von Nicht-Wort-Fehlern
- Korrektur des isolierten Wortes
- Kontextabhängige Wortkorrektur.

Nicht-Wort-Fehler, also Zeichenketten, die nicht in einer gegebenen Wortliste enthalten sind, können durch Pattern-Matching und N-Gramm-Analysetechniken erkannt werden. Die Korrektur eines isolierten Wortes wird, ohne die Einbeziehung von Informationen aus dem linguistischen oder textuellen Kontext in dem das Wort auftritt, durchgeführt. Anders bei der kontextabhängigen Wortkorrektur. Hier werden kontextabhängige Informationen genutzt, um Fehler zu erkennen, die selbst in ein gültiges Wort münden: (englisches Beispiel) from form.

### **Nicht-Wort-Fehler**

Für die Erkennung von Nicht-Wort-Fehlern können N-Gramme und Wörterbücher verwendet werden.

N-Gramme stellen eine Buchstaben oder Wortfolge aus N Elementen dar. Meist nimmt N die Größen 1 (Unigramm), 2 (Bigramm) oder 3 (Trigramm) ein. Für die Untersuchung von Texten wird eine N-Gramm-Tabelle entweder aus einem Corpus oder einem Wörterbuch erstellt. Die in der N-Gramm-Tabelle hinterlegten N-Gramme werden in dem zu untersuchendem String gesucht und ihre Häufigkeit mit einer Statistik verglichen. Werden N-Gramme gefunden, die sich stark von der Vergleichstatistik unterscheiden bzw. sehr selten sind, deutet dies auf einen Fehler hin. Ebenso deuten gefundene N-Gramme, die nicht in Wörtern vorkommen auf Fehler hin.

Der Wörterbuchansatz nutzt eine Liste von Wörtern, die als richtig geschrieben gelten. Wird ein Wort in einem String gefunden, das nicht in dem Wörterbuch auftaucht, ist es wahrscheinlich falsch geschrieben. Wie oben beschrieben, unterliegt ein Wörterbuch einem stetigen Wandel und muss regelmäßig aktualisiert werden. Je mehr ein Wörterbuch an eine spezielle Domäne angepasst ist (z.B. deutsche Sprache), desto besser funktioniert die Erkennung von Nicht-Wort-Fehlern. Zu große Wörterbücher erzeugen eine unakzeptabel hohe Anzahl von falschen Worten, die angenommen

werden. Aber auch zu kleine Wörterbücher sind problematisch, da viele richtige Wörter Fehlermeldungen erzeugen.

### **Korrektur des isolierten Wortes**

Nachdem Rechtschreibfehler in einem Text erkannt wurden, ist es wünschenswert diese auch zu korrigieren. Der nun beschriebene Ansatz nimmt als Ausgangspunkt ein einzelnes fehlerhaftes Wort. Für die Erstellung einer Liste mit Vorschlagsworten wird ein Wörterbuch herangezogen. Abhängig von der Größe des Wörterbuchs, seiner Abdeckung der deutschen Sprache und der Häufigkeit mit der neue Einträge vorgenommen werden, enthält die Liste entweder gute oder weniger gute Vorschlagsworte. Eventuell ist eine morphologische Verarbeitung eines Wortes notwendig, um passende Einträge im Wörterbuch zu finden. Eine weitere Möglichkeit für die Korrektur eines Wortes ist die Verwendung von Mustern (Pattern). Die Rechtschreibfehler werden hierbei nach typographischen, kognitiven und phonetischen Fehlern unterschieden. Für die Entwicklung einer nützlichen Rechtschreibkorrektur ist das aber nicht notwendig, da viele Techniken typographische und kognitive Fehler gleich gut behandeln. [Kuk92]

#### **Beispiel:**

- |                        |         |                     |            |
|------------------------|---------|---------------------|------------|
| • typografische Fehler | dei     | die, neune          | neuen      |
| • kognitive Fehler     | recieve | receive, conspiracy | conspiracy |
| • phonetische Fehler   | Baipass | Bypass, Cabare      | Cabaret    |

Der Korrekturprozess kann in drei Schritte unterteilt werden. Zuerst wird ein Fehler erkannt, für den im zweiten Schritt eine Liste mit Korrekturvorschlägen erstellt wird. Schritt drei ordnet diese Liste, wobei das richtige Wort möglichst weit vorne bzw. an erster Stelle der Liste erscheinen sollte. Der Sortierungsalgorithmus bezieht sich auf lexikalische Ähnlichkeit der Korrekturvorschläge zu dem fehlerhaften Wort oder einer geschätzten Wahrscheinlichkeit, mit der die Korrekturvorschläge dem fehlerhaften Wort entsprechen. In der Theorie kann die Liste der Vorschlagsworte aus dem gesamten Wörterbuch bestehen, wird aber für den praktischen Gebrauch auf wenige sinnvolle beschränkt. Der dritte Schritt kann weggelassen werden, wenn der Benutzer selbst die Auswahl des richtig geschriebenen Wortes aus den Korrekturvorschlägen vornimmt.

[Kuk92] unterteilt die Korrekturansätze in sechs Gruppen. Sie werden der Vollständigkeit halber genannt, aber nicht näher erläutert, da sie zu weit vom eigentlichen Thema der Arbeit wegführen würden.

- Minimum Edit Distance Techniques

- Similarity Key Techniques
- Rule-Based Techniques
- N-Gramm-based Techniques
- Probabilistic Techniques
- Neural Net Techniques.

### **Kontextabhängige Wortkorrektur**

Kontextabhängige Wortkorrektur ist für Real-Wort-Fehler, also Rechtschreibfehler die selbst wieder in einem gültigem Wort münden, und solche Fehler wichtig, die nur aus dem umliegenden Kontext eines Wortes erkannt werden können. Es reicht schon ein kleiner Tippfehler aus, um aus einer 'Farm' eine 'Form' zu machen. Isoliert betrachtet, ist das Wort korrekt geschrieben. Trotzdem stellt es, wenn man den ganzen Satz betrachtet, in dem es vorkommt, einen Fehler dar: 'In den Ferien war ich auf der Form meines Großvaters.' Kognitive, phonetische, syntaktische und grammatikalische Fehler können ebenfalls Real-Wort-Fehler erzeugen. Selbst das Fehlen ganzer Wörter oder die doppelte Verwendung sind nur durch die Einbeziehung des umliegenden Kontextes korrigierbar. Eine Vielzahl von Methoden, um das Problem der kontextabhängigen Wortkorrektur zu lösen, wurde in den letzten Jahren entwickelt. Oft spielt eine natürlich oder statistisch sprachliche Verarbeitung der Texte eine wichtige Rolle. [GR99] bietet eine chronologische Übersicht dieser Methoden und stellt selbst einen lernfähigen Algorithmus und eine Architektur mit Eigenschaften vor, welche versuchen, diese Art von Problemen zu lösen.

### **Zusammenfassung**

Für die Analyse von Texten, wie sie in dieser Arbeit untersucht wird, ist es sehr vorteilhaft, wenn sowohl die eingereichten Lösungen der Studenten, als auch die vordefinierte Vergleichslösung des Aufgabenerstellers frei von Rechtschreibfehlern sind. Es ist davon auszugehen, dass die Studenten vor der Einreichung ihrer Aufgaben keine Rechtschreibprüfung durch ein externes Programm (z.B. MS Word) vornehmen bzw. innerhalb einer Prüfungssituation keine Zeit dazu finden. Deshalb muss das CAA-Werkzeug eine Rechtschreibprüfung durchführen. Die besten Ergebnisse werden erzielt, wenn der Nutzer interaktiv die Korrektur vornimmt. Dafür muss der Editor in den der Antworttext eingetragen wird, über eine integrierte Rechtschreibprüfung verfügen oder in dem Fall, dass die Antwort als Textdatei hochgeladen wird, eine Rechtschreibprüfung der Datei vorgenommen und bei auftretenden Fehlern zu einer Korrektur aufgefordert werden. Ist nur eine automatische Korrektur des Textes möglich, wird es kompliziert. Schon bei Nicht-Wort-Fehlern kann nicht mit Sicherheit das richtige Wort bestimmt

werden, da wie oben beschrieben eine Liste mit möglichen Kandidaten erzeugt wird und theoretisch alle Wörter dieser Liste das gesuchte Wort darstellen können.

### 4.3. Aufgaben aus dem Bereich Mathematik

Aufgaben aus dem Bereich Mathematik sind auf jene Aufgaben zurückzuführen, die als Lösung eine Zahlen- oder Formeleingabe erwarten. In diesem Zusammenhang sind also auch Antworten auf physikalische, chemische oder anderweitige Aufgaben mit einbezogen. Die Antworten auf Fragen, die Zahlen- oder Formeleingaben als Lösung benötigen, stellen ebenfalls Freitextantworten dar und sind vergleichbar mit den in Kapitel 4.2 vorgestellten Kurzantworten. Meist ist die Antwort auf eine mathematische Frage kurz und prägnant. Ein Antwortsatz ist nicht zwingend erforderlich, ist aber sofern vorhanden, oft sehr einfach gehalten.

Im Folgenden werden Beispiele für Aufgaben gegeben, die eine Zahl- oder Formeleingabe als Antwort erwarten. Anschließend wird auf die Besonderheiten bei der Überprüfung der eingereichten Lösung nochmals eingegangen.

#### 4.3.1. Beispiele für mathematische Aufgaben

##### Zahl / Antwort als Satz

**Beispiel:**

Frage: Der Emscherdeich hat einen trapezförmigen Querschnitt. Die Deichkrone ist 5,6m breit. Der Deich hat eine Höhe von 3,9m. Der Böschungswinkel des Deiches beträgt  $30^\circ$ . Wie breit ist die Deichsohle s?

Musterlösung: 19,11m

Eingereichte Lösung 1: 19,1m

Eingereichte Lösung 2: 19,28m

Eingereichte Lösung 3: Die Dammsohle ist 19.11m breit.

Eingereichte Lösung 4: Sie ist 19.3m breit.

Für die Trennung zwischen Vor- und Nachkommastellen wurde sowohl der Punkt als auch das Komma verwendet. Die Lösungangaben schwanken, selbst wenn sie richtig sind, um einen gewissen Grad.

Für einen Abgleich mit der Musterlösung muss die Antwortzahl aus der eingereichten Lösung extrahiert werden. Über reguläre Ausdrücke lässt sich das am Besten bewerkstelligen. Als richtige Lösung sollte ein Intervall gewählt werden, das um das exakte Ergebnis angesiedelt ist.

## Formel

### Beispiel:

Frage: Differenzieren Sie den angegebenen analytischen Ausdruck:

$$y = 2 \sin \sqrt[3]{\frac{3}{x}}$$

Musterlösung:  $y' = -\frac{2 \cos \sqrt[3]{\frac{3}{x}}}{x^3 \sqrt[3]{9x}}$  für  $x > 0$

Eingereichte Lösung:  $y' = -\frac{2 \cos \sqrt[3]{\frac{3}{x}}}{x^3 \sqrt[3]{9x}}$  für  $x > 0$

Zur Beantwortung der Frage muss eine Formel eingegeben werden. Die Frage ist, wie das realisiert werden kann. Kapitel 4.3.3. beschäftigt sich mit diesem Thema.

### 4.3.2. Bewertung der eingereichten Lösung

Aufgaben die es erfordern mehrere Rechenschritte durchzuführen, um zum richtigen Ergebnis zu gelangen, können durch unterschiedliches Runden (falsch gerundet bzw. zu geringe Anzahl an Nachkommastellen für die Berechnung genutzt) in den Zwischenschritten ein leicht unterschiedliches Endergebnis erzielen. Wird das obige Beispiel angenommen, bei dem als Musterlösung der Wert 19,11 hinterlegt wurde, ist nur eine der vier eingereichten Lösungen als richtig zu bewerten. Die drei Lösungen, die nahe der Musterlösung liegen, sind wahrscheinlich durch Runden in den Zwischenschritten entstanden und sollten ebenfalls als richtig bewertet werden. Durch die Angabe eines Intervalls für die Musterlösung, der als richtige Lösung gilt, kann das Rundungsproblem umgangen werden. Der Lösungsbereich kann durch einen numerischen Wert oder eine Prozentangabe hinterlegt werden.



- Numerischer Wert: 19,11 +/- 0,2      18,91 – 19,31
- Prozentangabe: 19,11 +/- 1%      18,92 – 19,30

Durch die Definition mehrerer Bereiche können Antworten mit größerer Genauigkeit entsprechend honoriert werden.

Bei Aufgaben aus dem Bereich Physik oder Chemie werden oft Maßeinheiten verwendet, die für die Bewertung der Lösung wichtig sind. Fehlende Maßeinheiten werden oft mit Punktabzug bestraft. Da die Grundidee die Eingabe der Lösung in einen (Text-)Editor ist, muss die Maßeinheit mit angegeben werden. Für die automatische Kontrolle ist es einfacher, wenn in der Aufgabenstellung festgelegt wird, in welcher Maßeinheit die Lösung anzugeben ist. Zwar ist es nicht falsch die obige Strecke von 19,11m in Millimetern oder Dezimetern anzugeben, aber dann müsste das CAA-System jede eingereichte Lösung in die Maßeinheit der Musterlösung umrechnen, was unter Umständen schnell zu ungewollten Fehlern führen kann und eine Implementierung der verschiedensten Maßeinheiten für die Umrechnung erfordert.

Ein weiteres Problem sind Lösungswege und Folgefehler. Kontrolliert ein Lehrender manuell die Antworten der Studenten, so stehen ihm auch die Lösungswege zur Verfügung. Oft ist es so, dass die Abgabe der Lösungswege zwingend erforderlich ist, um im Zweifelsfall nachprüfen zu können, ob der Student die Lösung selbst berechnet hat oder nur die Lösung abschrieb. Zusätzlich kann der mit eingereichte Lösungsweg dazu dienen Folgefehler ausfindig zu machen und je nachdem wie die Bestimmungen sind, trotz falscher Lösung noch Punkte für eine Aufgabe zu erhalten. Wird in einem CAA-System nur die Lösung eingetragen, fällt dies alles weg. Es kann nur eine Aussage darüber getroffen werden, ob die eingereichte Lösung richtig oder falsch ist. Folgefehler werden nicht erkannt und die Erzeugung eines spezifischen Feedbacks fällt schwer.

Für die Bewertung von Formeln bietet es sich an, sie in ein passendes Format umzuwandeln und mittels Testdaten mit der Musterlösung zu vergleichen. Erhält man für Musterlösung und eingereichte Lösung dasselbe Ergebnis, kann die eingereichte Lösung als richtig bewertet werden. Auf diese Weise ist es auch möglich unterschiedliche Formeln als Lösung zuzulassen. Durch die Verwendung von Dezimalzahlen statt Brüchen oder unterschiedliche Klammerung und Schreibweise würde ein direkter Vergleich der Formeln sehr schwer fallen.

**Beispiel:**

$$\sqrt[a]{b^c} = b^{\frac{c}{a}}$$

### 4.3.3. Eingabe von Formeln bei der Frageerstellung und Lösungseingabe

Die Eingabe von Zahlen, die die Lösung für eine Aufgabe darstellen, lässt sich einfach handhaben. Das größte Problem ist die einheitliche Verwendung des Trennungszeichens für die Nachkommastellen: Komma oder Punkt. Die Verwendung sollte vom CAA-System klar angegeben werden, um Komplikationen zu vermeiden.

**Beispiel:**

Drei Millionen oder drei Euro?

3.000.000,000€ ↔ 3,000,000.000€

Für die Eingabe der Formeln bieten sich zwei Möglichkeiten an. Zum einen die direkte Eingabe in einen Texteditor (Textmodus) und zum anderen die Verwendung eines Formeleditors. Beide Möglichkeiten besitzen Vor- und Nachteile, die an einem kleinen Beispiel illustriert werden sollen:

| Direkte Eingabe in einen Texteditor             | Verwendung eines Formeleditors                             |
|-------------------------------------------------|------------------------------------------------------------|
| $y' = -(2 \cos((3/x)^{(1/3)}))/(x(9x^{(1/3)}))$ | $y' = -\frac{2 \cos \sqrt[3]{\frac{3}{x}}}{x^3 \sqrt{9x}}$ |

Die direkte Eingabe der Formel in den Texteditor erfolgt mittels einer speziellen Syntax, die erst erlernt werden muss. Nachdem die Syntax durch die Studenten erlernt und trainiert wurde, erfolgt die Eingabe einer Formel im Vergleich zur Eingabe über einen Formeleditor oft schneller. Problematisch ist, dass die Formel nicht sehr anschaulich ist und schnell Fehler in der Klammerung (fehlende oder zuviel Klammern) auftreten. Die Eingabe der Formel über den Formeleditor ist ohne große Einarbeitung meist intuitiv möglich. Durch die Auswahl der verschiedenen Sub-Formeln wird häufig eine höhere Zeitspanne für die Eingabe der Formel benötigt.

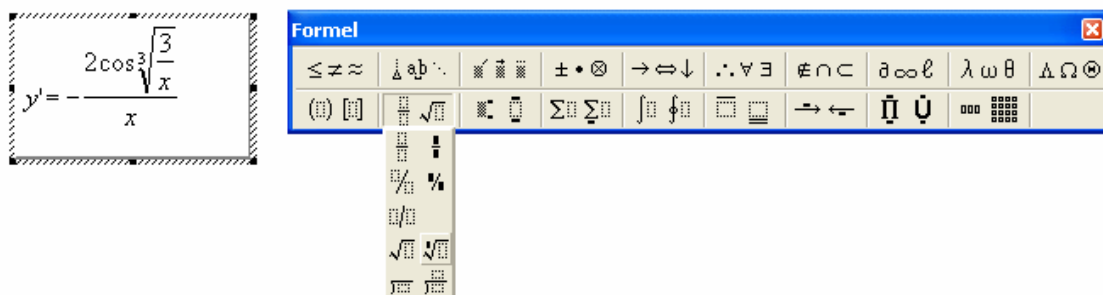


Abbildung 4-1 Formeleditor von MS Word

Der Vorteil eines Formeleditors ist, dass die Formel so aussieht, wie man sie sich vorstellt. Fehler in der Formel werden schneller erkannt und können einfach behoben werden.

Am besten wäre es, wenn die Syntax der eingegebenen Formeln vor der Einreichung geprüft wird. Sowohl für den Textmodus, als auch für den Formeleditor. Wenn die mittels Textmodus eingetragene Formel zur Überprüfung des Studenten, vor der Einreichung in eine grafische Ansicht der Formel umgewandelt wird, wäre das sehr hilfreich. Durch diese Hilfsmaßnahmen wird dem Studenten zwar ein Stück weit bei der Lösung der Aufgabe geholfen und ihm können keine Fehler durch fehlerhafte Klammerung oder falsche Benutzung von mathematischen Operatoren angelastet werden, aber die eingetragenen Formeln sind soweit fehlerfrei, dass sie weiter verarbeitet werden können.

Für die Kontrolle der eingereichten Lösung wäre die Umwandlung in eine gemeinsame Repräsentationsform praktisch. Wenn diese Repräsentationform zusätzlich noch gut austauschbar zwischen verschiedenen Programmen (auch externe Programme) ist und sich einfach speichern lässt, wäre das noch besser. MathML oder OpenMath wären geeignete Kandidaten.

MathML ist eine W3C (World Wide Web Consortium) Empfehlung und nutzt die XML-Spezifikation um mathematische Ausdrücke zu beschreiben. Die aktuelle Version ist 2.0. XML ermöglicht den einfachen Austausch von Daten zwischen verschiedenen Systemen. MathML-Daten können von Mathematikprogrammen berechnet werden (z.B. Mathematica [WR06]). Dadurch ist es möglich Aufgaben, die als Ergebnis eine Formel erwarten, über ein externes Programm berechnen zu lassen. Dazu müssen die Musterlösung, die Testdaten und die eingereichte Lösungen in MathML überführt und an das externe Programm zur Berechnung übermittelt werden. Ein Vergleich der Ergebnisse gibt Aufschluss über die Korrektheit einer eingereichten Lösung.

Die Beantwortung mathematischer Fragen in MathML-Syntax ist nicht praktikabel. Bei der Eingabe entstehen zu viele Fehler. Es ist ein XML-Editor notwendig, um ein halbwegs vernünftiges Ergebnis zu erzielen. Des Weiteren unterscheidet MathML zwischen Inhalts- und Darstellungssicht, die frei miteinander kombinierbar sind. Die Inhaltssicht beschreibt die Bedeutung des Ausdrucks und eignet sich besonders für die Verarbeitung der enthaltenen Daten. Die Daten werden dabei in der Prefix-Notation gespeichert. Die Darstellungssicht ist für die ‚optische‘ Beschreibung gedacht und speichert den mathematischen Ausdruck gewissermaßen so ab, wie man ihn selbst hinschreiben würde. Ein Beispiel soll dies verdeutlichen:  $(a + b)^2$

| Inhaltssicht                                                                                                                                                                                 | Darstellungssicht                                                                                                                                                                      |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> &lt;apply&gt;   &lt;power/&gt;   &lt;apply&gt;     &lt;plus/&gt;     &lt;ci&gt;a&lt;/ci&gt;     &lt;ci&gt;b&lt;/ci&gt;   &lt;/apply&gt;   &lt;cn&gt;2&lt;/cn&gt; &lt;/apply&gt; </pre> | <pre> &lt;msup&gt;   &lt;mfenced&gt;     &lt;mi&gt;a&lt;/mi&gt;     &lt;mo&gt;+&lt;/mo&gt;     &lt;mi&gt;b&lt;/mi&gt;   &lt;/mfenced&gt;   &lt;mn&gt;2&lt;/mn&gt; &lt;/msup&gt; </pre> |

#### 4.4. Richtlinien für die Beantwortung von Freitextaufgaben

Die automatische Verarbeitung von Freitextaufgaben kann sehr erleichtert werden, wenn eine Reihe von Richtlinien für ihre Beantwortung aufgestellt werden. Diese Richtlinien sollen einfach gehalten sein, damit die Studenten bei der Beantwortung der Fragen so wenig Zeit wie möglich mit der Einhaltung der Richtlinien zu kämpfen haben und sich auf ihre Prüfungsfragen konzentrieren können.

Freitext benötigt nicht viele Richtlinien. Für die automatische Analyse des Textes ist es vorteilhaft, wenn so wenige Abkürzungen wie möglich genutzt werden. Am Besten nur solche, die auch in Wörterbüchern (z.B. Duden) angegeben sind. Regeln für das Notieren von Stichpunkten wären sehr gut und könnten bei ihrer Auswertung weiterhelfen (bessere Erkennung der einzelnen Stichpunkte und Unterstichpunkt). Nur wird ein Student mit der Beantwortung einer Frage mehr zu tun haben, als noch zusätzlich spezielle Formatvorlagen einhalten zu müssen. Der Texteditor sollte über eine Rechtschreibkontrolle verfügen, die den Studenten auf Fehler aufmerksam macht und kleine Fehler, wie z.B. Buchstabendreher, automatisch korrigiert. Für den Fall, dass die eingereichte Lösung als Datei in das CAA-System hochgeladen wird, sollte eine sofortige Rechtschreibprüfung durchgeführt werden und ein Warnhinweis erfolgen, wenn zu viele Fehler aufgetaucht sind. Auf diese Weise hat der Einreichende die Möglichkeit die Rechtschreibfehler zu korrigieren und die Lösung nochmals abzugeben. Eine hohe Zahl an Rechtschreibfehlern führt zu einer schlechteren Bewertung / Einschätzung der eingereichten Lösung.

Für den mathematischen Bereich gilt es als erstes festzulegen, mit welchen Trennzeichen die Nachkommastellen separiert werden. Bei der Verwendung von Maßeinheiten sollten allgemein gültige Abkürzungen genutzt werden. Die Eingabe von Formeln im Textmodus erfordert die Aufstellung einer allgemein verbindlichen Syntax. Es bietet sich an, sich an gängigen Programmiersprachen zu orientieren und in das CAA-System eine Übersicht mit der zu benutzenden Syntax in die Hilfe zu integrieren.

## 4.5. Zusammenfassung

Mathematische Aufgaben wurden als eigenständiger Aufgabentyp betrachtet, da sich die Verarbeitung der eingereichten Lösungen von denen der Freitextaufgaben im Kern unterscheidet. Bei Freitextaufgaben werden die eingereichten Lösungen mit der Musterlösung verglichen und es wird versucht eine Aussage über die Richtigkeit der eingereichten Lösungen zu treffen. Für Kurzantwortfragen ist es meist möglich über Richtig und Falsch der Antwort zu entscheiden, da der Spielraum für ihre Beantwortung recht klein ist. Anders bei Aufsatzfragen oder Fragen die durch eine Reihe von Stichpunkten beantwortet werden. Hier ist die Aussage über die Richtigkeit der Lösung um vieles schwerer, wenn überhaupt machbar. Der Einsatz von Text Mining Verfahren gestaltet sich als schwierig.

Der Ansatz bei den mathematischen Aufgaben ist es dagegen, die eingereichten Lösungen und die Musterlösung in eine interne Repräsentation zu überführen und mittels Testdaten feststellen zu können, ob die eingereichten Lösungen richtig sind. Die Durchführung der Berechnung der Testdaten kann dabei über externe Programme erfolgen. Ist als Lösung keine Formel sondern eine oder mehrere Zahlen anzugeben, werden diese einfach verglichen. Um eine Toleranz für unterschiedliches Runden zu haben, wird ein Intervall um die genaue Lösung festgelegt. Die Angabe der Musterlösungen für mathematische Aufgaben gestaltet sich einfach. Für jede Variable muss eine Zahl angegeben werden. Durch das Ausrechnen der Aufgabe wird ein Ergebnis ermittelt, zu dem auch die Formel aus der eingereichten Lösung kommen muss, um als richtig zu gelten.

Ob für die Eingabe von mathematischen Daten der Textmodus oder der Formeleditor benutzt werden sollte, ist Geschmackssache. Beide Methoden haben ihre Vor- und Nachteile. Wenn oft Formeln eingegeben werden, wird die Eingabe der Formeln im Textmodus mit der Zeit schneller erfolgen. Falls die fertig eingegebene Formel auf Anfrage nochmals in grafischer Form zur besseren Überprüfung dargestellt wird, ist der Textmodus wahrscheinlich die beliebtere Eingabemethode.

Die in Kapitel 4.4. vorgestellten Richtlinien für die Beantwortung von Freitextantworten sind gewollt sehr kurz gehalten, um den Studenten größtmögliche Freiheit bei der Beantwortung der Aufgaben zu geben. Die Richtlinien können strenger sein und je nach Aufgabentyp anders gestaltet werden. Man sollte sich aber immer vor Augen halten, dass jede nicht eingehaltene Richtlinie Probleme bei der automatischen Kontrolle verursachen kann. In diesem Zusammenhang soll nicht unerwähnt bleiben, dass durch die Gestaltung der Fragen auf die Antworten eingewirkt werden kann.



# Kapitel 5

## Prototypische Implementierung

### 5.1. Programmumgebung

Der Prototyp wurde als Zusatzkomponente eines bereits existierenden Softwaresystems erstellt. Das Institut für Wissens- und Sprachverarbeitung der Otto-von-Guericke-Universität Magdeburg entwickelt e-learning-Komponenten als Erweiterung für das Content Management System (CMS) Plone. Plone setzt auf dem Anwendungsserver Zope auf und wurde als Basis gewählt, weil es bereits für die Web-Seite des Institut genutzt wird, quelloffen und plattformunabhängig (UNIX, LINUX, MacOS X, Windows) ist. Zope ist ebenfalls quelloffen und ebenso wie Plone in der Programmiersprache Python geschrieben. Die schon in das CMS eingespeisten Informationen (z.B. Kursmaterialien) bleiben nutzbar und die Erweiterungen von Plone können nahtlos in das vorhandene System integriert werden.

Erweiterungen werden in Plone ‚Produkte‘ genannt und sind, wie Plone selbst auch, in Python geschrieben. Drei Produkte wurden im Bereich e-learning erstellt und weiter entwickelt:

- LlsMultipleChoice
- ECAssignmentBox (ECAB)
- ECAutoAssessmentBox (ECAAB)

Mit ihrer Hilfe kann die Erstellung, Einrichtung, Verwaltung und Bewertung von Übungsaufgaben unterstützt werden.

LlsMultipleChoice dient zur Erstellung von Multiple Choice Fragen. Es können verschiedene Typen von Multiple Choice Fragen generiert werden. Eine vollständige Beschreibung der Features findet man in [APRa06].

Die ECAssignmentBox erlaubt das Erstellen, Einreichen und Benoten von Onlineaufgaben. Der Lehrende erstellt ein ECAssignmentBox-Objekt mit den Informationen über den Titel der Aufgabe, die Aufgabenstellung und den Einreichungszeitraum. Während des Einreichungszeitraums können die Studenten ihre Lösung online über die Plone-Web-Seite einreichen (als Datei hochladen oder direkt in ein Textfeld eintragen). Diese Einreichungen werden als ECAssignment-Objekte gespeichert und durchlaufen anschließend einen Workflow an dessen Ende die eingereichte Lösung entweder angenommen und benotet bzw. abgewiesen wird. Die

Bewertung und Benotung der Lösungen muss durch den Lehrenden erfolgen. Dieser kann alle Einreichungen einsehen und sie auch mit Kommentaren versehen.

ECAutoAssessmentBox stellt eine Erweiterung der ECAssignmentBox dar und ist von ihr abgeleitet. Mit der ECAutoAssessmentBox können Übungsaufgaben automatisch überprüft werden. Genutzt wird dies, um Programmieraufgaben automatisch zu kontrollieren. Die eigentliche Überprüfung wird von ECSpooler bereitgestellt. Der ECSpooler ist ein Webservice und verwaltet eine Warteschlange für die Einreichungen sowie verschiedene Backends. Die Backends führen die Kontrolle der Einreichungen durch. Sobald ein Backend mit der Prüfung einer Einreichung fertig ist, wird ein Feedback über das Ergebnis zurückgegeben und kann über die Plone-Web-Seite eingesehen werden. Der modulare Aufbau mit seiner genauen Trennung von Benutzeroberfläche, Spooler und Backend erlaubt eine einfache Erweiterung um neue Backends, die für die Kontrolle anderer Aufgaben zuständig sind. Der in diesem Kapitel vorgestellte Prototyp zur Textanalyse ist ein solches Backend. [APRa06], [APRb06]

## 5.2. ECAutoAssessmentBox (ECAAB)

### 5.2.1. Voraussetzungen für den Betrieb der ECAAB

Für den Betrieb der ECAutoAssessmentBox und des ECSpoolers sind eine Reihe von Bedingungen zu erfüllen. Zunächst werden die Programme Zope und Plone benötigt. Auf plone.org kann ein Installationspaket heruntergeladen werden, das alle notwendigen Programmpakete (Python, Plone, Zope) für den Betrieb einer Plone-Web-Seite bereitstellt. Nachdem die Plone-Web-Seite steht, können Produkte (Erweiterungen für Plone) in das ‚Product‘-Verzeichnis der Zope-Instanz kopiert und über das Zope Management Interface (ZMI) installiert werden. Benötigt werden die Produkte ECAssignmentBox, ECAutoAssessmentBox und ECSpooler. Weiterhin wird pysqlite, ein Python-Interface für die Benutzung von SQLite, und SQLite selbst benötigt.

Sobald alle Vorkehrungen getroffen sind, kann der ECSpooler gestartet und die Backends bei ihm angemeldet werden.

Für die Verwendung des Prototyps sind weitere Softwareinstallationen notwendig:

- NLTK-Lite [BLS06]  
Das Natural Language Toolkit ist eine Sammlung von Modulen, Datensammlungen und Tutorials, die das Erstellen von Programmen für die Computerlinguistik und natürlichsprachliche Verarbeitung unterstützt.
- PyEnchant [KP04]  
PyEnchant ist eine Bibliothek zur Rechtschreibprüfung für Python. Sie wird im Prototypen für die Erkennung und Korrektur von Rechtschreibfehlern eingesetzt.



- WordNet [MFT06]  
WordNet ist ein lexikalisches Referenzsystem. Substantive, Verben, Adjektive und Adverbien sind in Synonym-Sets organisiert, die jeweils ein lexikalisches Konzept präsentieren.
- PyWordNet [Ste04]  
PyWordNet stellt eine Schnittstelle zu WordNet zur Verfügung.
- Malaga [Beu05]  
Malaga kann in Verbindung mit Grammatiken für natürliche Sprache, zur morphologischen Analyse einzelner Wortformen verwendet werden. Die syntaktische Analyse ganzer Sätze oder Phrasen ist ebenfalls möglich. Aus Kompatibilitätsgründen zu PyMalaga, wurde mit der Version 6.8 gearbeitet.
- PyMalaga [Pio03]  
Dieses Python-Modul erlaubt das Verarbeiten von Malaga-Abfragen aus Pythonprogrammen.
- SQLite [Hip06]  
SQLite stellt eine einfach zu benutzende SQL-Datenbankengine für relationale Datenbanken bereit. Es existieren keine externen Abhängigkeiten, so dass SQLite schnell und unkompliziert einsetzbar ist. Datenbanken werden in einer einzelnen Datei gespeichert.
- PySqlite [Här06]  
Mit Hilfe von PySqlite ist die einfache Benutzung von SQLite in Python möglich.

### 5.2.2. Bedienung der ECAAB

Die Benutzung der ECAutoAssessmentBox unterscheidet sich nur in einigen Punkten von der Benutzung der ECAssignmentBox. Diese Punkte beziehen sich vor allem auf die Eingabe von Informationen, die für die automatische Korrektur durch die Backends des ECSpoolers durchgeführt werden. Um später statistische Informationen über die eingereichten Lösungen zu erhalten, muss die ECAutoAssessmentBox in ein ECFolder-Objekt eingebettet werden. Im folgenden Abschnitt wird die Erstellung eines Aufgabenblattes erklärt.

## Erstellung eines Aufgabenblattes

Die Eingabe der notwendigen Informationen unterteilt sich in zwei Bereiche. Im ersten Bereich sind allgemeine Angaben zu tätigen, die für jede ECAssignmentBox notwendig sind. Dazu gehören neben der Aufgabenstellung und der optionalen Angabe einer Antwortvorlage auch die Einstellung des Einreichungszeitraumes für die Aufgabe. Nur innerhalb der Einreichungsfrist ist es den Studenten möglich ihre Lösungen auf der Webseite des CAA-Systems einzureichen. Ganz wichtig ist die Auswahl des Backends am Ende des Formulars, das die automatische Überprüfung durchführen soll. Durch betätigen des Buttons ‚nächster‘, gelangt man zu den Einstellungen des ausgewählten Backends. Abhängig von der Arbeitsweise des Backends sind Einstellungsmöglichkeiten vorzunehmen. Meist bieten die Backends verschiedene Testarten an, aus denen die für Aufgabe geeignetste ausgesucht wird. Weiterhin ist die Angabe einer Modelllösung notwendig, mit deren Hilfe das Backend die Auswertung der eingereichten Lösungen vornimmt. Wie die Musterlösung auszusehen hat ist vom Backend und der Testart, die ausgewählt wurde, abhängig.

The screenshot shows a web interface for creating an assignment. At the top, there are tabs: 'inhalte', 'anzeigen', 'bearbeiten', 'eigenschaften', 'einreichungen', and 'zugriffsrechte'. The main title is 'Bearbeite Automatisch überprüfte Aufgabe'. Below it, there is a user information line: 'von admin --- Zuletzt verändert: 07.07.2006 09:56'. There are two links: '[default]' and '[backend]'. A description follows: 'Ermöglicht die Erstellung, Einreichung und Benotung automatisch überprüfter Aufgaben'. The 'Titel' field contains 'Aufgabe 1 - TextBackend'. The 'Beschreibung' field contains 'Allgemeinwissen (engl.)'. The 'Aufgabentext' field is a rich text editor with the text 'In which country is George W. Bush president?'. Below it is the 'Antwortvorlage' field, also a rich text editor. The 'Beginn der Einreichungsfrist' is set to 2006-07-01. The 'Ende der Einreichungsfrist' is set to 2006-07-10. There are checkboxes for 'Automatischer Zeilenumbruch im Antwortfeld' (checked) and 'Benachrichtigung bei Einreichungen' (unchecked). The 'Backend' dropdown is set to 'Text (1.0)'. At the bottom, there are buttons for 'nächster', 'speichern', and 'abbrachen'.

Abbildung 5-1 Erstes Formular zur Aufgabenerstellung

Die Backends führen die eigentliche automatische Überprüfung der eingereichten Lösungen durch. Sie sind auf einen Aufgabentyp spezialisiert. So stehen z.B. Backends für die Kontrolle von Programmieraufgaben unter Python und Haskell zur Verfügung. Programmieraufgaben werden überprüft, indem die studentische Lösung und die vorgegebene Musterlösung mit verschiedenen Testdaten ausgeführt und verglichen wird. Eine Syntaxprüfung findet ebenfalls statt.

Abbildung 5-2 Zweites Formular zur Aufgabenerstellen – Informationen zum Backend

## Einreichung einer Lösung

Damit Studenten eine Lösung für eine ECAutoAssessmentBox-Aufgabe einreichen können, müssen sie sich bei der Plone-Web-Seite anmelden und die entsprechende Aufgabe auswählen. Ist eine Einreichung möglich, kann der Student seine Lösung entweder in ein Textfeld eintragen oder direkt eine Datei mit der Lösung hochladen.

Abbildung 5-3 Einreichung einer Lösung

Sobald der Student seine eingegebenen Daten übermittelt, wird eine Überprüfung der Lösung vorgenommen und eine automatische Rückmeldung über den Status der Aufgabe zurückgegeben.

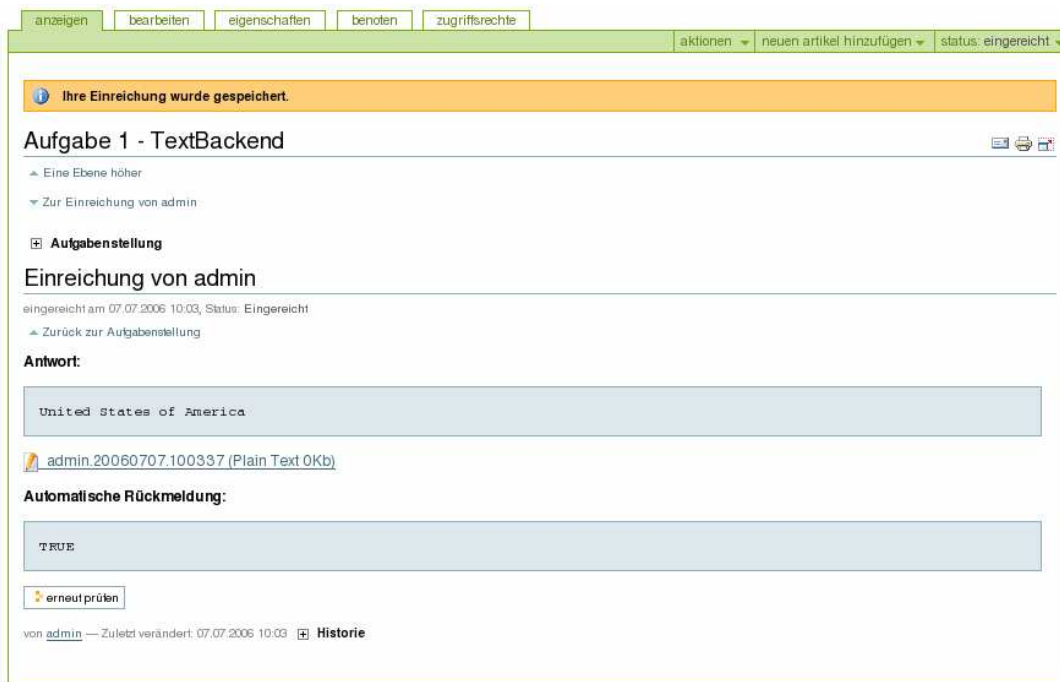


Abbildung 5-4

Die Rückmeldung enthält Informationen über die Richtigkeit der eingereichten Lösung bzw. aufgetretene Fehler während der Verarbeitung. Der Inhalt der Rückmeldung wird innerhalb des Backends festgelegt

Mehrfacheinreichungen sind möglich. Bereits eingereichte Lösungen bleiben dabei erhalten und werden mit dem Status ‚Ersetzt‘ markiert.

## Prüfen und Bewerten

Die endgültige Entscheidung über die Akzeptierung einer eingereichten Lösung erfolgt durch den Lehrenden. Dieser kann bei Bedarf eine nochmalige Überprüfung der Aufgabe vornehmen, falls z.B. die Testdaten oder das Backend selbst geändert wurden.

## 5.3. Der Prototyp – TextBackend

Das implementierte Backend erlaubt es Textantworten zu analysieren. Es unterscheidet die eingereichten Texte zwischen Kurzantworten und Freitexten. Kurzantworten

bestehen nur aus einer Wortgruppe oder einem kurzen Satz. Längere Antworten werden als Freitextantworten angesehen. Das Backend wird im folgenden TextBackend genannt.

Ebenso wie bei den bisher existierenden Backends des ECSpoolers, wird bei der Erstellung der Aufgabe eine Musterlösung in der ECAAB hinterlegt. Durch Analyse der Musterlösung und der eingereichten Lösung des Studenten wird eine Aussage über die ‚Richtigkeit‘ der Lösung getroffen. Ziel ist es nicht, ähnlich wie bei Programmieraufgaben, eine exakte Aussage über ‚Falsch‘ und ‚Richtig‘ treffen zu können. Vielmehr ist hier das Ergebnis der Analyse als Unterstützung für den Lehrenden zu sehen, der einen schnellen Überblick über die eingereichten Lösungen haben möchte. Als direkte Ergebniswerte könnte z.B. der Anteil der Wörter einer in der Musterlösung hinterlegten Stichwortliste ermittelt werden. Die Möglichkeiten sind sehr weit. Im Folgenden wird auf einige dieser Möglichkeiten eingegangen, welche Probleme dabei auftreten und wie sie gelöst werden können.

### **5.3.1. Die Rechtschreibung**

In Kapitel 4 wurde auf die Grundlagen der computergestützten und teilweise automatischen Rechtschreibprüfung für Texte eingegangen. Für die Überprüfung einer Aufgabe ist es vorteilhaft, wenn sowohl die eingereichten Lösungen als auch die Musterlösungen frei von Rechtschreibfehlern sind.

#### **PyEnchant**

Das TextBackend setzt die Python-Bibliothek PyEnchant für die Rechtschreibprüfung ein. PyEnchant basiert auf der Enchant-Bibliothek von Dom Lachowicz. [KP04]

PyEnchant kann auf verschiedene im Hintergrund arbeitende Rechtschreibprogramme, Provider genannt, zugreifen. Zur Auswahl stehen: myspell, aspell und ispell. Die Wahl des zu nutzenden Providers kann manuell eingestellt werden.

Um PyEnchant für die Rechtschreibprüfung benutzen zu können, muss ein ‚Dict‘-Objekt angelegt werden. Dieses repräsentiert ein Wörterbuch und kann benutzt werden, um eine Rechtschreibprüfung einzelner Wörter bzw. einen Vorschlag für ein falsch geschriebenes Wort zu erhalten. Auf diese Weise können nur Nicht-Wort-Fehler erkannt werden. Für ein fehlerhaftes Wort kann eine Liste von Vorschlagswörtern abgefragt werden. Das gesuchte Wort befindet sich nicht zwingend am Anfang der Liste bzw. kann auch gar nicht enthalten sein, wenn der Rechtschreibfehler zu gravierend ist.

**Beispiel:**

```
>>> import enchant
>>> d= enchant.Dict("de_DE")
>>> d.check('Programinterfaze')
False
>>> d.suggest('Programinterfaze')
['Programminterfaces', 'Programminterface',
'Programmiererfahrung', 'Programmunterlagen',
'Programmiertechnik', 'Programmierzeit', 'Programmierkosten',
'Programmierfehlers', 'Programmierfehler', 'Programmzeitung']
>>>
```

Das gesuchte Wort im obigen Beispiel ist ‚Programminterface‘. Die fehlerhafte Schreibweise wurde erkannt. Die Liste der Vorschlagswörter enthält das richtige Wort an zweiter Stelle. Soll eine automatische Korrektur des fehlerhaften Wortes durchgeführt werden, muss die Vorschlagsliste so umsortiert werden, dass das richtige Wort am Anfang der Liste steht. Das fehlerhafte Wort und das gesuchte richtige Wort sind sich oft sehr ähnlich. Mit Hilfe der Levenshtein-Distanz und des Dice-Koeffizienten kann die Ähnlichkeit zwischen ihnen berechnet werden. Das Vorschlagswort mit der größten Ähnlichkeit wird als das gesuchte richtige Wort angenommen.

**Levenshtein-Distanz**

Die Levenshtein-Distanz gibt die minimale Anzahl von Operationen an, die notwendig sind, um eine Zeichenkette in eine andere zu überführen. Zulässige Operationen sind: Einfügen, Löschen und das Ersetzen eines einzelnen Zeichens durch ein Anderes.

**Beispiel:**

Programinterfaze    Programminterface

Notwendige Änderungsoperationen:

- Einfügen von ‚m‘
- Ersetzen von ‚z‘ durch ‚c‘

Die Levenshtein-Distanz beträgt 2.

Für obiges Beispiel ergeben sich folgende Levenshtein-Distanzen:

**Tabelle 5-1 Levenshtein-Distanz – Berechnung für eine Vorschlagsliste**

| Zu untersuchendes Wort | Levenshtein-Distanz zu ‚Programminterface‘ |
|------------------------|--------------------------------------------|
| Programminterface      | 2                                          |
| Programminterfases     | 3                                          |
| Programmunterlagen     | 5                                          |
| Programmierfehler      | 7                                          |
| Programmierzeit        | 7                                          |
| Programmiererfahrung   | 8                                          |
| Programmierfehlers     | 8                                          |
| Programmierkosten      | 8                                          |
| Programmiertechnik     | 8                                          |
| Programmzeitung        | 9                                          |

Die Wahl des Wortes mit der geringsten Levenshtein-Distanz ist das gesuchte Wort.

### Dice-Koeffizient

Der Dice-Koeffizient ist eine Möglichkeit mittels N-Grammen Wortähnlichkeiten zu berechnen. Der Dice-Koeffizient zweier Wörter berechnet sich nach der Formel:

$$d(a,b) = \frac{2|T(a) \cap T(b)|}{|T(a)| + |T(b)|}$$

T(x) ist die Zerlegung des Wortes x in seine N-Grammen. Das Ergebnis der Berechnung ist ein Wert zwischen 0 und 1 und gibt den Grad der Ähnlichkeit an, wobei höhere Werte eine höhere Ähnlichkeit angeben.

**Beispiel:**

Berechnung des Dice-Koeffizienten unter Verwendung von Trigrammen.

| Programinterfaze                                                           | Programminterface                                                               |
|----------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| Pro, rog, ogr, gra, ram, ami,<br>min, int, nte, ter, erf, rfa,<br>faz, aze | Pro, rog, ogr, gra, ram, amm,<br>mmi, min, int, nte, ter, erf,<br>rfa, fac, ace |

$$d(\text{Programinterfaze}, \text{Programminterface}) = \frac{2 \cdot 11}{14 + 15} = \frac{22}{29} = 0,7586$$

Berechnet man die Dice-Koeffizienten für die im obigen Beispiel zurückgegebenen Vorschlagsworte und das als fehlerhaft erkannte Wort ‚Programinterfaze‘, so ergeben sich die Werte:

**Tabelle 5-2 Dice-Koeffizient – Berechnung einer Vorschlagsliste**

|                      | Bi-Gramm | Tri-Gramm |
|----------------------|----------|-----------|
| Programminterface    | 0,84     | 0,76      |
| Programminterfaces   | 0,81     | 0,73      |
| Programmierzeit      | 0,62     | 0,37      |
| Programmiererfahrung | 0,59     | 0,44      |
| Programmierkosten    | 0,58     | 0,34      |
| Programmierfehler    | 0,58     | 0,41      |
| Programmiertechnik   | 0,56     | 0,33      |
| Programmierfehlers   | 0,56     | 0,40      |
| Programmunterlagen   | 0,56     | 0,47      |
| Programmzeitung      | 0,48     | 0,37      |

Beide N-Gramme sehen in dem Wort ‚Programminterface‘ die größte Ähnlichkeit.

### **Evaluierung der Levenshtein-Distanz und des Dice-Koeffizient bei der Berechnung des besten Vorschlagswortes**

Die Evaluierung erfolgt mit Testdaten, die der Homepage [AK04] entliehen sind. Sie werden dort genutzt, um das Leistungsvermögen neuerer aspell-Versionen zu demonstrieren. Die Testdaten bestehen aus einer Liste mit 547 Wörtern, die einmal richtig und einmal falsch geschrieben wurden.

Auszug aus den Testdaten:

```
Accosinly  Occasionally
Ciculer    Circler
Circue     Circle
Maddness   Madness
Occasionally Occasionally
...        ...
```

Für den Test werden nur die beiden Provider myspell und aspell betrachtet. Myspell ist der Standardprovider für Windows und aspell der für Linux. Zuerst wurde getestet, wie viele der 547 Wörter von den Providern erkannt wurden, wenn sie richtig geschrieben sind. Das gibt Aufschluss darüber, wie gut eine Sprache durch den Provider abgedeckt wird. Myspell ist mit einer Abdeckung von 98,2% ein wenig besser als aspell mit 96,9%.

Jetzt folgt die interessante Frage, wie gut sich die Levenshtein-Distanz und der Dice-Koeffizient eignen, an Hand eines fehlerhaft geschriebenen Wortes das richtige Wort aus einer Liste von Korrekturvorschlägen zu identifizieren.



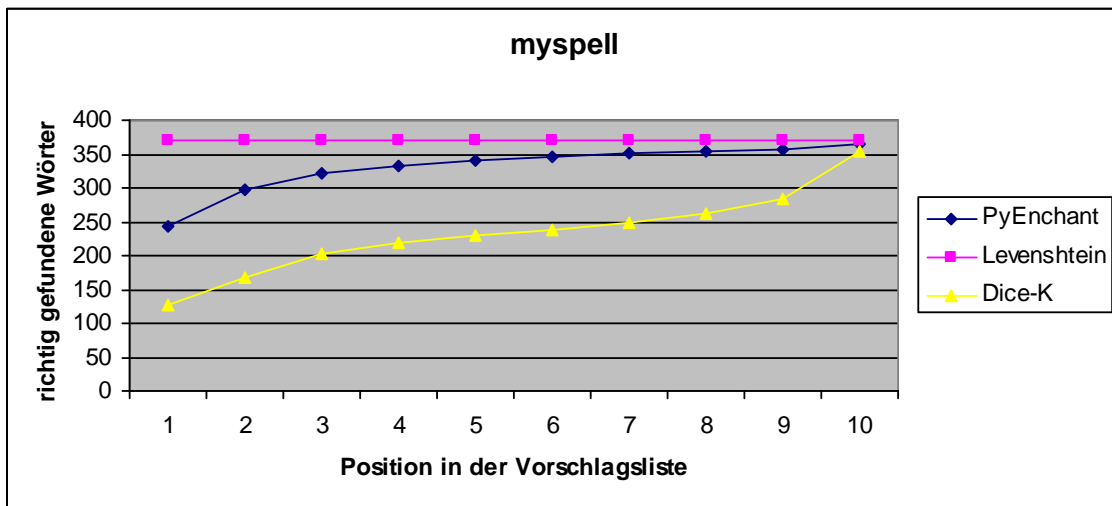


Abbildung 5-5myspell-Evaluierung

Die Levenshtein-Distanz positioniert nach der Sortierung immer das gesuchte Wort an die erste Stelle der Vorschlagsliste. Die Erfolgsrate für die Korrektur des fehlerhaften Wortes mit dem ersten Wort der Vorschlagsliste liegt für die Levenshtein-Distanz bei 67%. PyEnchant und der Dice-Koeffizient starten mit 44% und 23% Wahrscheinlichkeit, dass das erste Wort der von ihnen sortieren Liste der Vorschlagswörter gleich das gesuchte ist.

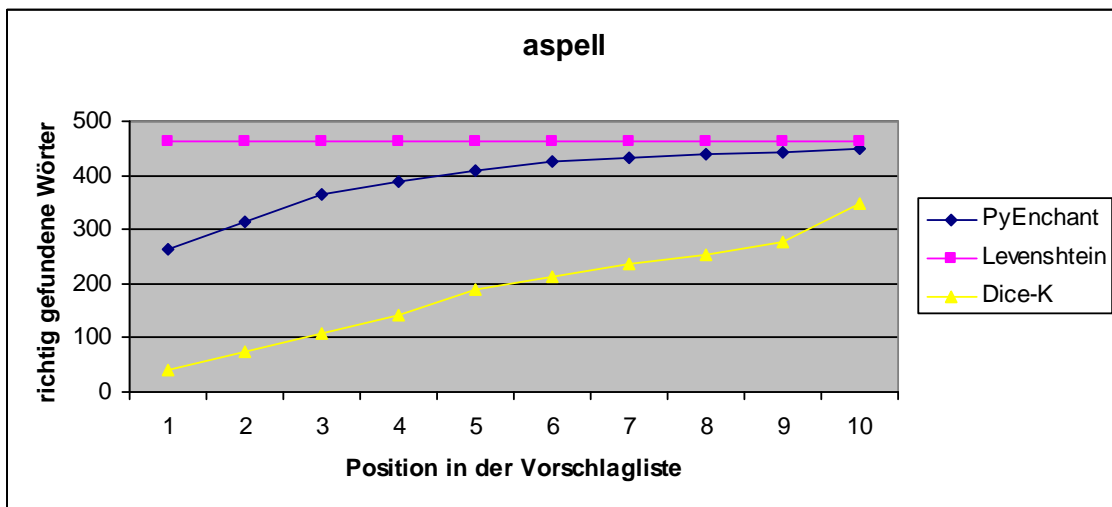


Abbildung 5-6aspell-Evaluierung

Bei der Verwendung von aspell liegt die Trefferquote der Levenshtein-Distanz bei 85%. PyEnchant positioniert zu 48% das gesuchte Wort auf den ersten Platz der Vorschlagsliste. Der Dice-Koeffizient liefert ein extrem schlechtes Ergebnis, eine 7%-Chance für einen Treffer beim ersten Wort der Liste.

Die von PyEnchant gelieferte Vorschlagsliste enthält das richtige Korrekturwort innerhalb der ersten zehn Worte zu 67% für myspell und zu 85% für aspell. Damit ist aspell trotz schlechterem Ergebnis beim ersten Test, der eindeutige Favorit und sollte

als Provider für PyEnchant genutzt werden. Auch wenn die Liste der Vorschlagswörter das richtige Korrekturwort enthält, ist es nicht automatisch erkennbar, welche Position es innerhalb der von PyEnchant herausgegebenen Vorschlagsliste einnimmt. Für die automatische Korrektur der fehlerhaften Wörter wäre es am Besten, wenn das gesuchte Wort zu Beginn der Vorschlagsliste steht. Die Levenshtein-Distanz leistet dafür hervorragende Dienste und sollte entsprechend eingesetzt werden.

## **Zusammenfassung**

PyEnchant ist ein einfach zu benutzendes Python-Modul für die Rechtschreibprüfung. Leider ist es nur möglich Nicht-Wort-Fehler zu erkennen, für diese können aber Korrekturvorschläge eingeholt werden. Sowohl die Levenshtein-Distanz, als auch die Dice-Koeffizient sind hilfreich, um aus einer Liste von Vorschlagsworten für ein fehlerhaft geschriebenes Wort möglichst das gesuchte Wort zu identifizieren. Die Levenshtein-Distanz liefert dabei bessere Ergebnisse, als der Dice-Koeffizient. Die Gründe liegende Ähnlichkeit ist aber kein Garant dafür, dass wirklich das gesuchte Wort gefunden wird. Mögliche Fallstricke sind z.B., dass das gesuchte Wort nicht im Wörterbuch enthalten ist oder die Verfälschung zu stark ist.

### **5.3.2. Die Vorverarbeitung des Textes**

Um eine Aussage über die Richtigkeit einer eingereichten Lösung vornehmen zu können, wird ein Vergleich mit einer Musterlösung durchgeführt. Für den Vergleich ist es notwendig die beiden Texte in eine geeignete Repräsentation zu überführen, deren Grundlage Indexterme bilden. Die Erstellung der Indexterme kann zeichenketten- oder wortbasiert erfolgen. Alle Indexterme eines Textes können als Vektor zusammengefasst werden, der für jeden Indexterm die Häufigkeit innerhalb des Textes angibt.

Der zeichenkettenbasierte Ansatz verwendet N-Gramme. Jedes N-Gramm stellt einen Indexterm dar. Der größte Vorteil bei der Verwendung von N-Grammen ist die Unabhängigkeit von der im Text verwendeten Sprache und dessen Themengebiet. Des Weiteren ist die Verarbeitung sehr schnell und Rechtschreibfehler sind von geringerer Bedeutung. Nachteilig ist die Ungenauigkeit, die durch das Zerlegen der Wörter entsteht. Die Anzahl der N-Gramme eines Textes übersteigt meist die Anzahl der enthaltenen Wörter und somit werden zu Vektoren zusammengefasste N-Gramme sehr lang.

Wird ein wortbasierter Ansatz gewählt, stellen die Wörter selbst die Indexterme dar. Daher ist es notwendig Wortgrenzen bestimmen zu können, um die Wörter eines Textes voneinander trennen zu können. Die getrennten Wörter werden auch als Token bezeichnet.

Die Vorverarbeitung eines Textes erfolgt in drei Schritten:

1. Zerlegung von Zeichenketten in Token
2. Entfernung von Stoppwörtern
3. Stammformreduzierung

### Zerlegung von Zeichenketten in Token

Die Zerlegung der Zeichenketten in Token kann auf verschiedene Weisen erfolgen. Die einfachste Lösung wäre anzunehmen, dass Wörter durch Leerzeichen getrennt sind und diese als Trennsymbol benutzt werden. Ein großes Problem dieser Methode ist, dass entstehende Token oft Satzzeichen enthalten und in späteren Verarbeitungsschritten nicht mehr verwendbar sind.

*Du trägst sehr leicht, wenn du nichts hast;  
Aber Reichtum ist eine leichtere Last. (J. W. von Goethe)*

```
['Du', 'trägst', 'sehr', 'leicht,', 'wenn', 'du',  
'nichts', 'hast;', 'Aber', 'Reichtum', 'ist', 'eine',  
'leichtere', 'Last.']
```

Schon in diesem kurzen Textbeispiel sind drei Token mit Satzzeichen enthalten.

Eine bessere Lösung ist die Verwendung von regulären Ausdrücken, um die Wörter aus dem Text herauszulösen. Reguläre Ausdrücke beschreiben Suchmuster mit Hilfe syntaktischer Regeln. Zeichenketten die einem gesuchten Suchmuster entsprechen, werden extrahiert und in unserem Beispiel zu einer Liste zusammengefasst. Eine kurze Einführung in die Möglichkeiten regulärer Ausdrücke findet sich im Anhang.

Für das obige Beispiel wurde als Suchmuster eine Folge von alphanumerischen Zeichen [a-zA-Z0-9] bestimmt, die mindestens die Länge eins hat. Der reguläre Ausdruck für Python ist `r'\w+'`.

```
['Du', 'trägst', 'sehr', 'leicht', 'wenn', 'du',  
'nichts', 'hast', 'Aber', 'Reichtum', 'ist', 'eine',  
'leichtere', 'Last']
```

Schon durch einen relativ einfachen regulären Ausdruck wurden die störenden Satzzeichen weggelassen. Komplexere reguläre Ausdrücke erlauben das Erkennen von Uhrzeiten, Zahlen- oder Datumsangaben und vieles mehr. Oft muss der reguläre Ausdruck an die entsprechende Situation angepasst werden, wie z.B. für die Erkennung der unterschiedlichen Verwendung von Komma und Punkt bei Zahlenangaben in verschiedenen Ländern.

Die Verarbeitung von regulären Ausdrücken unter Python kann entweder direkt mit dem integrierten `re`-Modul oder denen im Natural Language Toolkit (NLTK) bereitgestellten Tokenizern durchführen werden. Die Benutzung des NLTK ist komfortabler. [Wei05], [BLS06]

## Entfernung von Stoppwörtern

Nachdem ein Text in Token zerlegt wurde, ist es nützlich die Wörter auszusortieren, die häufig in Texten vorkommen, aber nichts über den Inhalt aussagen( Artikel, Pronomen, Präpositionen). Zu diesem Zweck werden Stoppwortlisten benutzt, in denen diese Wörter gespeichert sind. Das TextBackend nutzt als Stoppwortliste eine einfache Textdatei, in der jede Zeile ein Stoppwort enthält. Auf diese Weise kann die Liste einfach editiert und in Token umgewandelt werden.

Abhängig vom Ziel der Textanalyse und der vom Text behandelnden Domäne, kann der Inhalt der Stoppwortliste stark variieren.

Benutzt man den ‚bag of word‘-Ansatz, bei dem die Reihenfolge der Wörter innerhalb des Textes vernachlässigbar ist, können über Mengeoperationen mit der Token-Liste des Textes und der Token-Liste der Stoppwörter auf einfache Weise die Stoppwörter entfernen werden:

```
['Last', 'Reichtum', 'trägst', 'leicht', 'leichtere',  
'hast', 'Du']
```

Ist die Reihenfolge der innerhalb der Token-Liste wichtig, ist ein größerer Rechenaufwand zu betreiben, da für jedes Wort der Liste einzeln geprüft werden muss, ob es sich um ein Stoppwort handelt:

```
['Du', 'trägst', 'leicht', 'hast', 'Reichtum',  
'leichtere', 'Last']
```

## Stammformreduzierung

Die aus der Tokenisierung gewonnenen und anschließend von Stoppwörtern gesäuberten Wortlisten enthalten Wörter in verschiedenen Flexionsformen. Durch die Stammformreduzierung werden verschiedene morphologische Varianten eines Wortes auf eine gemeinsame Stammform zurückgeführt. Die Reduzierung auf die Stammform kann über einen Stemming-Algorithmus oder dem Nachschlagen in einem Wörterbuch erfolgen.

Einer der bekanntesten Stemming-Algorithmen ist der Porter-Stemmer. Er ist nach seinem Autor Martin Porter benannt und wurde 1980 veröffentlicht. Der Algorithmus verarbeitet typische morphologische und flektierte Endungen von Wörtern und versucht durch deren schrittweise Abtrennung die Stammform eines Wortes zu erzeugen. Die erste Ausgabe des Porter-Stemmers war auf die englische Sprache ausgelegt. Inzwischen gibt es beim Snowball-Projekt [Por06] auch einen Porter-Stemmer für die deutsche Sprache. Weitere Stemming-Algorithmen sind Lovins und Paice/Husk.

Die vom Porter-Stemmer entfernten Suffixe unterscheiden sich für jede Sprache. In der deutschen Sprache sind das z.B. -en, -est für Verben, -lich und -ig für Adjektive und -heit, -keit, -ung für Substantive. Diese Liste ist nicht vollständig und soll nur einen Einblick gewähren. Beispiele für durch Porter gestemmtete deutsche Wörter sind in der folgenden Tabelle abgebildet.

**Tabelle 5-3 Stemming von Wörtern**

| Wort                 | Stammform       | entfernte Suffixe |
|----------------------|-----------------|-------------------|
| aufeinanderfolge     | aufeinanderfolg | -e                |
| aufeinanderfolgen    | aufeinanderfolg | -en               |
| aufeinanderfolgend   | aufeinanderfolg | -end              |
| aufeinanderfolgende  | aufeinanderfolg | -e, -end          |
| aufeinanderfolgenden | aufeinanderfolg | -en, -end         |
| aufeinanderfolgender | aufeinanderfolg | -er, -end         |

Die andere Möglichkeit die Stammform eines Wortes zu erhalten, ist die Benutzung eines Wörterbuches in dem für jedes Wort seine Stammform hinterlegt ist. Die Verwendung einer formalen Grammatik für natürliche Sprachen kann genutzt werden, um ein Wort zu analysieren. Wenn in dem Programm ‚Malaga‘ ein Wort morphologisch untersucht wird, wird unter anderem auch die Stammform eines Wortes zurück. Über das PyMalaga-Modul ist auch der Zugriff direkt aus Pythonprogrammen realisierbar.

```
['Du', 'trag', 'leicht', 'hast', 'Reichtum', 'leicht',
'Last']
```

## Zusammenfassung

Nachdem die Vorverarbeitung eines Textes abgeschlossen und der Text damit in eine einheitlichere Form gebracht wurde, können Untersuchungen an ihm vorgenommen werden. Die aus einem Text gewonnen Wortlisten werden teilweise weiterverarbeitet, um einen Vergleich einer eingereichten Lösung und der dazugehörigen Musterlösung durchführen zu können. Im nachfolgenden Kapitel wird auf Analysen zur Bestimmung der Korrektheit einer eingereichten Lösung eingegangen.

### 5.3.3. Implementierte Testarten des Prototypen

Der Prototyp stellt ein Backend der ECAAB dar und ist auf die Analyse und Auswertung von Testeingaben spezialisiert. Das Backend trägt die Bezeichnung TextBackend.

Die Einstellungen zum TextBackend werden bei der Erstellung eines Aufgabenblattes (siehe 5.2.2.) festgelegt. Jede Variante die eingereichten Antworten zu verarbeiten wird als eine Testart bezeichnet. Zusätzlich zu der Testart, der eine Aufgabe unterzogen wird, muss die Sprache ausgewählt werden, in der die Aufgabe beantwortet werden soll. Dies wurde so umgesetzt, dass die gewünschte Testart jeweils in der Sprache im Auswahlfeld selektiert werden muss, mit der die Analyse durchführen soll:

|                   |                                 |
|-------------------|---------------------------------|
| <b>Beispiele:</b> |                                 |
| •                 | ‚Einfache Antwort‘      deutsch |
| •                 | ‚Simple Answer‘      englisch   |

Die Musterlösung muss ebenfalls in der entsprechenden Sprache vorliegen. Nicht jede in 4.2. und 4.3. vorgestellten Verfeinerungsformen der Freitextantworten lässt sich mit jeder Testart überprüfen. Das liegt daran, dass unterschiedliche Analysen vorgenommen werden und diese nicht für jede Testart sinnvolle Ergebnisse liefern. Ein langer Text z.B., der als ein einzelnes Wort behandelt wird, wird keine sinnvollen Ergebnisse liefern.

Die Einstellungen für die verwendete Sprache beeinflusst die Wahl des Wörterbuches, welches für das Rechtschreibmodul (PyEnchant) genutzt wird. Des Weiteren wird für die Stammformreduzierung der zu der verwendeten Sprache passende Stemmer geladen bzw. bei der Verwendung eines Wörterbuches für die Stammformreduktion eine sprachabhängige Grammatik geladen (PyMalaga).

Ist die eingereichte Lösung ein Freitext längeren Umfangs, so ist es möglich die genutzte Sprache automatisch zu erkennen. Realisierbar ist dies, in dem für die ersten n Worte eines Textes (ca. 50 ist ein guter Wert) eine Rechtschreibprüfung in verschiedenen Sprachen durchgeführt wird. Die Sprache mit der die wenigstens Rechtschreibfehler gefunden werden, entspricht der Sprache des Textes. Diese Art der Spracherkennung ist natürlich auf die Anzahl der Sprachen beschränkt, die durch das Rechtschreibprüfprogramm abgedeckt sind. Für einen Vergleich der Musterlösung mit der eingereichten Lösung ist es zwingend erforderlich, dass beide in der gleichen Sprache vorliegen. Aus diesem Grund wurde auf eine automatische Spracherkennung im Prototypen verzichtet.

Das Feedback für die einzelnen Testarten ist nicht über die Einstellungen des Backends bei der Aufgabenerstellung definierbar (siehe 5.2.2.). In der jetzigen Version ist die Art

des Feedbacks im Backend verankert und muss direkt im Quellcode geändert werden, wenn es anders gestaltet werden soll. Im Standardfall gibt der Prototyp als Feedback an, ob die Aufgabe richtig oder falsch oder teilweise (Prozentangabe) gelöst wurde. Zusätzlich wird bei einigen Testarten die eingereichte Lösung mit angegeben. Die Rückgabe der richtigen Lösung ist als Feedback nicht geeignet, da Mehrfacheinreichungen möglich sind und auf diese Weise die Studenten die Möglichkeit besitzen, die richtige Lösung ohne Arbeitsaufwand per ‚copy & paste‘ einzureichen.

## Kurzantwort

Besteht die eingereichte Antwort nur aus einem einzelnen Wort oder einer Gruppe von Wörtern, die eine Bedeutungseinheit bilden, ist die Vorverarbeitung der Antwort nicht notwendig. Für die Prüfung auf Korrektheit der eingereichten Lösung werden Wörter, die eine Bedeutungseinheit bilden, als ein Token aufgefasst. Im TextBackend werden Fragen, die ein einzelnes Wort oder eine Gruppe zusammengehöriger Worte als Antwort erwarten, als ‚Einfache Antworten‘ (eine Testart des Backends) bezeichnet. Als Antworten können theoretisch jegliche alphanumerische Zeichenketten verlangt werden. Das können Worte, Zahlen oder sonstige Zeichenketten sein. Die Musterlösung für die gesuchte Antwort besteht aus einem Wort bzw. eine Gruppe von Wörtern. Die Prüfung auf Korrektheit der eingereichten Antwort kann als String-Vergleich durchgeführt werden. Durch einen String-Vergleich kann aber nur eine Lösung, die 1:1 der Musterlösung entspricht, als richtig gewertet werden.

| <b>Beispiel:</b>             |                                   |
|------------------------------|-----------------------------------|
| <b>Einzelne Zeichenkette</b> | <b>Bedeutungseinheit</b>          |
| Frankreich                   | Vereinigte Staaten von Amerika    |
| 1776                         | double median(double x, double y) |
| X:common                     | Albert Einstein                   |

Wird als Antwort eine natürliche Sprache gewählt, ist es möglich auftretende Rechtschreibfehler zu erkennen und gegebenenfalls zu korrigieren. Die eingereichte Lösung kann auf diese Weise trotz fehlerhafter Orthographie als richtig gewertet werden. Das PyEnchant-Modul bietet die Fehlererkennung von Nicht-Wort-Fehlern an und kann eine Liste von Vorschlagswörtern für die Korrektur eines falsch geschriebenen Wortes anbieten. Um eine eingereichte orthographisch fehlerhafte Lösung mit der Musterlösung zu vergleichen, wird mittels PyEnchant das falsch geschriebene Wort identifiziert und eine Liste mit Korrekturvorschlägen erzeugt. Die Levenshtein-Distanz hilft das Wort mit der größten Ähnlichkeit zum fehlerhaft geschriebenen Wort aus der Liste der Vorschlagswörter zu identifizieren. In 5.2.3. wurden gute Ergebnisse bei der Verwendung der Levenshtein-Distanz zur Bestimmung des gesuchten, richtig geschriebenen Wortes aus einer Liste mit Vorschlagsworten

nachgewiesen. Nachdem das fehlerhafte Wort ersetzt wurde, wird ein nochmaliger Vergleich mit der Musterlösung durchgeführt.

Ist die eingereichte Lösung trotz Rechtschreibkorrektur nicht korrekt, wurde eventuell ein Synonym der Musterlösung als Antwort eingereicht. Mittels WordNet, für das ein Schnittstellenmodul zu Python (PyWordNet) existiert, können Synonyme für Wörter bestimmt werden. Eine Prüfung der Synonyme mit der eingereichten Antwort ergibt Klarheit und führt möglicherweise dazu, dass die eingereichte Antwort als richtig bewertet wird, wenn sie mit einem der Synonyme übereinstimmt. Für das TextBackend wurde nur die kostenfreie englische Fassung von WordNet genutzt. GermaNet, die kostenpflichtige deutsche Variante von WordNet und kann bei Bedarf ohne großen Aufwand mit eingebunden werden.

**Beispiel:**

Frage: Welches Land trat 1941 in den zweiten Weltkrieg ein, nachdem Japan Pearl Harbor überfiel?

Musterlösung: USA

Eingereichte Lösung: Unitet States of America

Die im Beispiel gestellte Frage würde von einem menschlichen Prüfer als richtig gewertet werden, mit dem Hinweis, dass ‚Unitet‘ fehlerhaft geschrieben wurde. Die Testart ‚Einfache Antworten‘ kommt zu dem gleichen Ergebnis, allerdings mit der Einschränkung, dass nur Synonyme englischer Wörter gefunden werden können. Das Ergebnis der Prüfung einer eingereichten Antwort ist entweder, dass die Antwort als richtig oder falsch beantwortet gilt.

Eine weitere Testart, die dem Bereich Kurzantwort zugeordnet ist, nimmt als eingereichte Lösungen Sätze oder Wortgruppen entgegen, die ein in der Musterlösung vordefiniertes Lösungswort bzw. eine Wortgruppe enthalten. Die Bezeichnung lautet ‚Einfache Antwortsätze‘. Eine Tokenisierung und Stammformreduzierung wird nicht vorgenommen. Für diese Testart wird die in der Musterlösung hinterlegte Zeichenkette in der eingereichten Lösung gesucht und sofern sie gefunden wurde, die Lösung als richtig bewertet. Ist das Ergebnis der Überprüfung negativ, wird eine Rechtschreibprüfung der eingereichten Lösung durchgeführt und eine erneute Kontrolle eingeleitet. Sollte das Ergebnis wieder negativ ausfallen, werden die Synonyme der Musterlösung in der eingereichten Lösung gesucht.



**Beispiel:**

Frage: Welches Land trat 1941 in den zweiten Weltkrieg ein, nachdem Japan Pearl Harbor überfiel?

Musterlösung: USA

Eingereichte Lösung: Die Unitet States of America traten 1941 in den zweiten Weltkrieg ein.

Das Beispiel zeigt eine eingereichte Lösung, in der das gesuchte Lösungswort zum einen durch ein Synonym ausgedrückt und zum anderen auch noch fehlerhaft geschrieben wurde.

Eine weitere Variante von Aufgaben, bei denen kurze Texte (oft nur ein einzelnes Wort) als Antwort gesucht werden, sind Lückentexte. Aufgabe ist es die fehlenden Wörter im Text mit den richtigen Wörtern zu füllen. Die gesuchten Wörter werden, nachdem sie aus dem Text extrahiert wurden, wie ‚Einfache Antworten‘ behandelt.

**Mathematik**

Für Aufgaben bei denen als Ergebnis eine Zahl erwartet wird, wurde die separate Testart ‚*Mathematikantwort*‘ implementiert. Ihre Besonderheit gegenüber ‚Einfache Antworten‘, bei denen die Ergebniszahl einhundertprozentig mit der in der Musterlösung hinterlegten Zahl übereinstimmen muss, ist, dass ein Intervall definierbar ist, in dem die eingereichte Lösung liegen kann, um als richtig bewertet zu werden. Das Intervall kann mit einem numerischen Wert (nw) oder einem prozentualen Wert (pw) um die Musterlösung angegeben werden. Über reguläre Ausdrücke können Antwortsätze ausgewertet werden. Für eine Aufgabe wird nur ein Ergebnis erwartet.

**Beispiel:**

Eingereichte Antwort: Der Lastenaufzug muss 13,6m entfernt sein

Musterlösung: 13,5m

Mögliche Intervallangaben:

- $nw = 0,2$  Eine Wert zwischen 13,3 – 13,7 gilt als richtige Lösung.
- $pw = 1,5$  Ein Wert zwischen 13,2975 – 13,7025 gilt als richtige Lösung

**Freitext**

Nachdem das TextBackend Fragen, die eine kurze Antwort erwarten, einigermaßen abgedeckt hat, bleibt noch die Königsdisziplin übrig: Freitexte. Gemeint sind Texte mit mehr als einem Satz oder einer Wortgruppe. Für die Analysen werden die Texte einer Vorverarbeitung unterzogen, wie sie in 5.3.2. beschrieben ist.

Eine Testart für Freitexte untersucht, ob bestimmte Stichworte in der eingereichten Lösung vorkommen. Die Liste der gesuchten Stichworte ist in der Musterlösung hinterlegt und spiegelt die wichtigsten Daten wieder, die in einer eingereichten Lösung enthalten sein sollten. Die Reihenfolge der Wörter im Text ist dabei unerheblich und ermöglicht es, die aus der Vorverarbeitung gewonnenen Token über einfache Mengenoperationen mit der in der Musterlösung gespeicherten Stichworte von den restlichen Token zu trennen. Der Anteil der Stichworte, die in der eingereichten Lösung gefunden wurden, wird als Feedback zurückgegeben. Eine Andere Variante den Text auf die Stichworte zu prüfen, nutzt den eingereichten Text direkt ohne Vorverarbeitung. Die einzelnen Stichworte werden einfach im per Stringsuche im Text gesucht. Der Vorteil liegt darin, dass Bezeichnungen wie ‚Vereinigte Staaten von America‘ erkannt werden, die sonst über die Tokenisierung verloren gegangen wären. Besonders für Sprachen, die keinen so einfachen Wortaufbau zu Komposita besitzen wie die deutsche Sprache, ist das wichtig.

Die Auswertung, die aus der automatischen Analyse von Freitexten gewonnen wird, ist abhängig von der Aufgabenstellung mehr oder weniger befriedigend. Ist die Aufgabenstellung ein Abruf von Fakten, so kann über Schlüsselwörter eine gute Aussage über die Richtigkeit getroffen werden. Sind in der Aufgabenstellung dagegen Erörterungen, Interpretationen oder ähnliches gefordert, so hilft die Analyse von Schlüsselwörtern nicht besonders weiter. In diesen Fällen ist es notwendig, dass der Lehrende (Kontrollleur) die eingereichte Antwort manuell überprüft und bewertet.

## Statistische Untersuchungen

Statistische Untersuchungen eines Textes sind als eigene Testart implementiert. Der Grund dafür ist, dass diese Untersuchungen für den Lehrenden bestimmt sind. Die Ergebnisse der statistischen Analyse könnten auch in das reguläre Feedback, das die Studenten erhalten, mit eingearbeitet werden, birgt aber nicht unbedingt einen Vorteil für den Studenten. Dieser könnte die Ergebnisse falsch interpretieren und nachfolgende Einreichungen unter Gesichtspunkten seiner, eventuell falschen, Interpretation der statistischen Untersuchungen durchführen. Dem Lehrenden dagegen bieten sie einen schnellen Überblick über verschiedene Aspekte eines Textes. Besonders der Vergleich der Analyseergebnisse über alle eingereichten Antworten kann neue Erkenntnisse mit sich bringen. Der Prototyp kann nur einzelne Texte statistische auswerten, eine Vergleichsmöglichkeit mit anderen eingereichten Texten ist nicht integriert.

Als Analyseergebnisse stehen folgende Informationen zur Verfügung:

- Anzahl der Zeichen / Wörter / Sätze
- Durchschnittliche Wort- und Satzlänge
- Häufigkeit der einzelnen Worte / N-Gramme (mit und ohne Stoppwörtern)
- Anteil der Rechtschreibfehler (Nicht-Wort-Fehler)
- Erkennung der im Text benutzen Sprache

Ein gutes Beispiel für den Nutzen, den schon das Zählen der Wörter im Text haben kann, ist die Kontrolle, ob die Studenten einen ausreichend langen Text abgegeben haben. Die Aufgabenstellung einer einzureichenden Aufgabe könnte den Passus enthalten, dass mit der Abgabe von ungefähr fünf Textseiten gerechnet wird. Da pro Seite ungefähr 300 Wörter geschrieben werden, wird von der eingereichten Lösung erwartet, dass sie ca. 1500 Wörter enthält. Ist die Abweichung zu groß, kann der Student darauf aufmerksam gemacht oder die Benotung des Textes beeinflusst werden. Die Entscheidung liegt am Ende beim Lehrenden.

## Zusammenfassung

Das TextBackend bietet verschiedene Testarten für die Analyse von Freitextantworten an. Eine vollständige Analyse eines Textes, insbesondere sind damit aufsatzartige Texte gemeint, mit der Aussage, ob sie eine gestellte Frage richtig beantworten, ist nicht möglich. Kurzantworten können ausgewertet werden. Durch die Erkennung von Synonymen und einer Rechtschreibprüfung sind auch Antworten möglich, die nicht 1:1 mit der Musterlösung übereinstimmen. Durch die speziellere Testart ‚Mathematikantwort‘ werden kleinere Abweichungen in der Berechnung des Ergebnisses einer Aufgabe abgefangen. Die Überprüfung eines Textes nach Stichworten

bietet dem Lehrenden die Möglichkeit schnell zu erkennen, ob eine eingereichte Lösung wichtige Begriffe die in ihr enthalten sein sollten beinhaltet. Nimmt man diese Informationen, um eine Vorsortierung der eingereichten Lösungen vorzunehmen, kann der Lehrende unterstützt werden, in dem er besonders aus dem Rahmen fallende Antworten zuerst begutachtet.

# Kapitel 6

## Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit einem speziellen Aspekt von CAA-Systemen. In CAA-Systemen eingesetzte Tests besitzen die Eigenschaft, dass sie automatisch kontrolliert und bewertet werden können. Das liegt an der klaren Definition, welche der Antwortmöglichkeiten einer Frage richtig und welche falsch sind. Die Informationen über einen Test mit der Angabe der richtigen und falschen Antworten werden in geeigneten Datenstrukturen gespeichert, z.B. IMS QTI. Auswahlantworttests sind die besten Beispiele für solche Tests. Für einen Auswahlantworttest ist eine kleine Menge an möglichen Antworten vorgegeben, von denen aber nur eine richtig ist. Die Aufgabe des Studenten ist es, die richtige Antwort auszusuchen. Mit Hilfe der hinterlegten Information kann die Aufgabe automatisch ausgewertet werden.

Meist bieten CAA-Systeme (siehe 3.) auch Fragen an, die als Antwort einen Eintrag in ein Textfeld verlangen. Die geforderten Antworten bestehen oft nur aus einem Wort und müssen genau dem entsprechen, was in einer Musterlösung hinterlegt ist, um als richtig bewertet zu werden. Häufig werden diese Fragen in Form von Lückentexten benutzt. Besteht die Möglichkeit einen langen Text als Antwort einzugeben, wird oft darauf verwiesen, dass diese Aufgabe manuell durch den Lehrenden nachkontrolliert und bewertet werden muss. In einigen Fällen können durch den Lehrenden Schlüsselwörter definiert werden, die in der eingereichten Lösung gesucht werden.

In Kapitel 4 wurde eine Unterscheidung von Texteinreichungen vorgenommen. Abhängig von der Aufgabenstellung sind verschiedene Antworttypen möglich. Der separaten Betrachtung von Mathematikaufgaben lag die Überlegung zu Grunde, dass eingetragene mathematische Antworten, sofern sie nicht schon selbst die Lösung der Aufgabe darstellen, in MathML umgewandelt und an externe Programme zur Berechnung übermittelt werden können. Dadurch können Formeln, die von der Musterlösung abweichen an Hand von Testdaten überprüft werden.

Fragen die als Antwort die Eingabe von Freitext, also die freie Formulierung der Antwort, ermöglichen, unterscheiden sich von den herkömmlich in CAA-Systemen eingesetzten Testarten. Die Definition ihrer Musterlösung mit einer einfachen Abgrenzung der falschen und richtigen Antworten ist nicht leicht, wenn überhaupt durchführbar. Für Mathematikaufgaben und Fragen, die eine Kurzantwort fordern, ist die Definition einer Musterlösung möglich. Die Durchführung einer Rechtschreibprüfung und das Finden von Synonymen eines in der Musterlösung hinterlegten Lösungswortes ermöglichen letztendlich die automatische Korrektur der eingereichten Lösung fernab eines einfachen Stringvergleiches. Die Korrektur eines Aufsatzes ist dagegen nur in einem begrenzten Maße möglich. Das Computerprogramm, das die Analyse des Textes durchführt, müsste gewissermaßen

verstehen, worum es in dem Text geht und anschließend bewerten, ob die Frage beantwortet wurde. Da dies nicht so einfach möglich ist, wird auf die Suche von Schlüsselbegriffen, Ähnlichkeitsuntersuchungen zwischen Dokumenten und statistische Analysen zurückgegriffen.

Das in Kapitel 5 vorgestellte TextBackend zeigt beispielhaft, wie eine Auswertung von Freitext vorgenommen werden kann. Die Grundidee ist dabei immer, dass für eine gestellte Aufgabe eine Musterlösung existiert, die für die Kontrolle der eingereichten Lösungen herangezogen wird. Die Bewertung einfacher, auch mathematischer Antworten, ist möglich. Für die Analyse von langen Textpassagen werden Schlüsselwörter innerhalb des Textes gesucht. Weiterhin kann der Text statistisch untersucht werden. Die endgültige Bewertung eines langen Freitextes (Aufsatz oder Essay) muss immer noch durch den Lehrenden durchgeführt werden. Trotz dieser Einschränkung ist es das Ziel des TextBackends Informationen für den Lehrenden aus der eingereichten Lösung zu gewinnen, die ihm bei der Auswertung und Bewertung der eingereichten Lösungen helfen können.

Die Einschränkung, dass für die Überprüfung der eingereichten Lösung nur die Musterlösung zur Verfügung steht, bringt sowohl Vor- als auch Nachteile mit sich. Auf der einen Seite ist der Lehrende sehr frei in der Aufgabenerstellung. Jegliche ihm einfallende Aufgabe kann er erstellen. Für die Überprüfung braucht er nur eine Musterlösung hinterlegen. Bei Kurzantwortfragen stellt dies kein Problem dar, allerdings könnte die Einbeziehung zusätzlicher Informationen große Vorteile bringen. Werden neben der Musterlösung weitere Informationen benötigt, um eine Überprüfung der eingereichten Lösungen vorzunehmen, müssen diese erst einmal beschafft und in eine geeignete Form gebracht werden. Werden eingereichte Antworten z.B. mit einer Dokumentensammlung verglichen, könnte die Zuordnung der Antwort zu einer bestimmten Kategorie innerhalb der Dokumentensammlung etwas über den Wahrheitsgehalt der Antwort aussagen. Je feiner die Granularität der Kategorien ist, desto genauer ist das Ergebnis. Wird dagegen eine grobe Granularität genutzt, desto allgemeingültiger können die Fragen beantwortet werden.

Das TextBackend kann in späteren Arbeiten um viele weitere Funktionen ergänzt werden. Die Einbeziehung zusätzlicher Informationen für die Analyse von Aufsatzfragen ist ein Beispiel. Zusätzlich ist die Integrierung eines eigenständigen Backends für mathematische Aufgaben in die ECAAB denkbar. Da das TextBackend eine Reihe externer Module benutzt, ist mit einer Verbesserung der Ergebnisse der Analyseergebnisse zu rechnen, wenn neue verbesserte Versionen von ihnen erscheinen. Die von PyEnchant benutzten Provider würden z.B. mit erweiterten Wörterbüchern mehr Wörter erkennen (z.B. Fachtermini) und entsprechend auch besser Vorschlagswörter für fehlerhafte Wörter erzeugen. Die Probleme mit Rechtschreibfehlern können außerdem über die Integrierung eines Editors, der eine Rechtschreibprüfung beinhaltet, in das CAA-System abgemildert werden.

# Literaturverzeichnis

- [All95] ALLEN, James: Natural Language Understanding. 2te Auflage. The Benjamin/Cummings Publishing Company, Inc., 1995. – ISBN 0–80530334–0
- [APRa06] AMELUNG, Mario; PIOTROWSKI, Michael; RÖSNER, Dietmar: EduComponents: Experiences in E-Assessment in Computer Science Education. In ITiCSE '06: Proceedings of the 11th annual conference on Innovation and technology in computer science education, ACM Press
- [APRb06] AMELUNG, Mario; PIOTROWSKI, Michael; RÖSNER, Dietmar: Elektronische Einreichung von Übungsaufgaben: Implementierung, Einsatz und Erfahrungen, Otto-von-Guericke-Universität Magdeburg – Wissensbasierte Systeme und Dokumentenverarbeitung, 2006
- [AK04] ATKINSON, Kevin: GNU Aspell, 2004 – Online im Internet: URL: <http://aspell.sourceforge.net> (10.08.2006)
- [Beu05] BEUTEL, Björn: Malaga – Online im Internet: URL: <http://www.malaga-grammars.de> (offline, im Google-Cache auf dem Stand vom 10.07.2005 einsehbar)
- [Bla06] Blackboard Academic Suite™. Blackboard Inc. – Online im Internet: URL: <http://blackboard.com> (10.08.2006)
- [BLS06] BIRD, Steven; LOPER, Edward; SPEER, Rob: Natural Language Toolkit – Online im Internet: URL: <http://nltk.sourceforge.net> (10.08.2006)
- [BM04] BULL, Joanna ; MCKENNA, Colleen: Blueprint for Computer-Assisted Assessment. 1te Auflage. Routledge Falmer, 2004. – ISBN 0–415–28704–9
- [Bro06] Diploma Campus™/ EDU Campus™. Brownstone Research Group, Inc. – Online im Internet: URL: <http://www.brownstone.net> (10.08.2006)
- [Dal05] DALHEIMER, Matthias K.: Latex - kurz und gut. 2te Auflage. O'Reilly, 2005. – ISBN 3–89721–500–4
- [Duden96] SCHOLZ-STUBENRECHT, Werner; WERMKE, Matthias: DUDEN – Die deutsche Rechtschreibung. 21te Auflage. Dudenverlag, 1996. – ISBN 3–411–04011–4

- [Eul01] EULER, Timm: Informationsextraktion durch Zusammenfassung maschinell selektierter Textsegmente, Universität Dortmund, Diplomarbeit, 2001
- [Fri03] Friedl, Jeffrey E. F.: Reguläre Ausdrücke. 2te Auflage. O'Reilly Verlag, 2003. – ISBN 3897213494
- [GNetz] G-Netz, Das Gesundheitsnetzwerk – Online im Internet: URL: [http://www.g-netz.de/Der\\_Mensch/skelett/index.shtml](http://www.g-netz.de/Der_Mensch/skelett/index.shtml) (10.08.2006)
- [GP] GLOW-PROJECT – Online im Internet: URL: <http://www.glow.ac.uk> (10.08.2006)
- [GR99] Golding, Andrew R.; ROTH, Dan: A Winnow-Based Approach to Context-Sensitive Spelling Correction
- [Hal06] Hot Potatoes™. Half-Baked Software Inc. – Online im Internet: URL: <http://hotpot.uvic.ca> (10.08.2006)
- [Här06] HÄRING, Gerhard: PySqlite, 2006 – Online im Internet: URL: <http://initd.org/tracker/pysqlite> (10.08.2006)
- [Hip06] HIPPI, Richard: SQLite – Online im Internet: URL: <http://www.sqlite.org> (10.08.2006)
- [HS91] HESSE, Jürgen ; SCHRADER, Hans C.: Testaufgaben - Das Übungs-Programm. 1te Auflage. Eichborn Verlag, 1991. – ISBN 3–8218–1219–2
- [ILI06] ILIAS™. ILIAS open source – Online im Internet: URL: <http://www.ilias.de> (10.08.2006)
- [IMS] IMS Global Learning Consortium – Online im Internet: URL: <http://www.imsproject.org> (10.08.2006)
- [KP04] KELL, Ryan: PyEnchant, 2004 – Online im Internet: URL: <http://pyenchant.sourceforge.net> (10.08.2006)
- [Kni02] KNIGHT, Peter T.: Summative Assessment in Higher Education: practices in disarray. In: Studies in Higher Education. Routledge, part of the Taylor & Francis Group, 2002, S. 27(3):275–286
- [Kuk92] KUKICH, Karen: Techniques for Automatically Correcting Words in Text. In: ACM Computing Surveys (CSUR). ACM Press, 1992, S. 377–439



- [Lea] E LEARNING@ED. Computer Aided Assessment. – Online im Internet: URL: <http://www.elearn.malts.ed.ac.uk/services/CAA/index.phtml> (10.08.2006)
- [MFT06] MILLER, George A.; FELLBAUM, Christiane; TENGI, Randee; WOLFF, Susanne; WAKEFIELD, Pamela; LANGONE, Helen; HASKELL, Benjamin: WordNet, Princeton University – Online im Internet: URL: <http://wordnet.princeton.edu> (10.08.2006)
- [Pio03] PIOTROWSKI, Michael: PyMalaga, 2003 – Online im Internet: URL: <http://www.dynalabs.de/mxp/python/index.html> (10.08.2006)
- [PKP03] PRINZ, Peter ; KIRCH-PRINZ, Ulla: C++ Das Übungsbuch. 1te Auflage. mitp, 2003. – ISBN 3–8266–0998–0
- [PR05] PIOTROWSKI, Michael; RÖSNER, Dietmar: Integration von E-Assessment und Content-Management. In DeLFI2005: 3. Deutsche e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V., Bonn, GI-Verlag. (Lecture Notes in Informatics (LNI) no. None)
- [Que06] Questionmark™ Perception™. Questionmark Computing Limited – Online im Internet: URL: <http://www.questionmark.com> (10.08.2006)
- [Sch06] SCHNEIDER, Wolf: Deutsch für Kenner. 2te Auflage. Piper Verlag GmbH, 2006. – ISBN 3–492–24461–0
- [SG94] SCHÄFER, Wolfgang ; GEORGI, Kurt: Mathematik-Vorkurs. 2te Auflage. B.G. Teubner Verlagsgesellschaft, 1994. – ISBN 3–8154–2078–4
- [Sta06] STANGL, Werner: [werner.stangl]s arbeitsblätter – Lernziele, 2006 – Online im Internet: URL: <http://arbeitsblaetter.stangl-taller.at/LERNZIELE> (10.08.2006)
- [Por06] PORTER, Martin: Snowball, 2006 – Online im Internet: URL: <http://snowball.tartarus.org> (10.08.2006)
- [Ste04] STEELE, Oliver: PyWordNet, 2004 – Online im Internet: URL: <http://osteele.com/projects/pywordnet> (10.08.2006)
- [TML] Tutorial Modelling Language – Online im Internet: URL: <http://www.ilrt.bris.ac.uk/netquest/> (10.08.2006), Seite funktionierte nur noch teilweise
- [UniBe] Universität Bern, Institut für Wirtschaftsinformatik – Online im Internet: URL: <http://www.ie.iwi.unibe.ch> (10.08.2006)

- [UniNH] University of New Hampshire, Blackboard QuickTips for Instructors – Online im Internet: URL: <http://www.unh.edu/blackboard/quicktips/>. (10.08.2006)
- [UniUI] University of Ulster, Computer Assisted Assessment – Online im Internet: URL: <http://www.science.ulster.ac.uk/caa> (10.08.2006)
- [Wei05] Weigend, Michael: Python – Ge-Packt. 2te Auflage. mitp-Verlag, 2005. – ISBN 3-8266-1512-3
- [WR06] Wolfram Research, Inc.: Mathematica – Online im Internet: URL: <http://www.wolfram.com> (10.08.2006)

# Abbildungsverzeichnis

|                                                                                           |    |
|-------------------------------------------------------------------------------------------|----|
| Abbildung 4-1 Formeleditor von MS Word .....                                              | 42 |
| Abbildung 5-1 Erstes Formular zur Aufgabenerstellung .....                                | 50 |
| Abbildung 5-2 Zweites Formular zur Aufgabenerstellen – Informationen zum Backend<br>..... | 51 |
| Abbildung 5-3 Einreichung einer Lösung .....                                              | 51 |
| Abbildung 5-4.....                                                                        | 52 |
| Abbildung 5-5myspell-Evaluierung .....                                                    | 57 |
| Abbildung 5-6aspell-Evaluierung.....                                                      | 57 |

# Tabellenverzeichnis

|                                                                                                  |    |
|--------------------------------------------------------------------------------------------------|----|
| Tabelle 2-1 Behauptung/Grund – Varianten der Beantwortung.....                                   | 21 |
| Tabelle 4-1 verwendete Abkürzungen in Freitexten .....                                           | 31 |
| Tabelle 5-1 Levenshtein-Distanz – Berechnung für eine Vorschlagsliste .....                      | 55 |
| Tabelle 5-2 Dice-Koeffizient – Berechnung einer Vorschlagsliste.....                             | 56 |
| Tabelle 5-3 Stemming von Wörtern .....                                                           | 61 |
| Tabelle 0-1 kognitive Lernziele [BM04].....                                                      | 77 |
| Tabelle 0-2 Die wichtigsten Sonderzeichen und –sequenzen in regulären Ausdrücken<br>[Wei05]..... | 78 |

# Anhang

## Blooms Taxonomie von Lernzielen

Bloom unterschied Lernziele in drei Dimensionen: kognitive, affektive und psychomotorische Lernziele. Für die Einteilung der in Kapitel 2.3. vorgestellten Fragetypen sind kognitiven Lernziele interessant. [Sta06]

**Tabelle 0-1 kognitive Lernziele [BM04]**

| <b>Kompetenz</b> | <b>Demonstrierte Fähigkeiten</b>                                                                                                                   |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| Wissen           | Abruf von Informationen<br>Wissen von Fakten, Zeitangaben, Ereignissen, Orten                                                                      |
| Verstehen        | Interpretation von Informationen in eigenen Worten<br>Den Sinn begreifen                                                                           |
| Anwendung        | Die Anwendung von Methoden, Theorien und Konzepten an neuen Situationen                                                                            |
| Analyse          | Identifizierung von Elementen<br>Die Fähigkeit die Verbindungen, Wechselwirkungen und Beziehungen zwischen den Elementen verstehen                 |
| Synthese         | Abstrahieren von gegebenem Wissen<br>Alte Ideen benutzen, um neue zu kreieren<br>Wissen aus verschiedenen Bereichen organisieren und zuordnen      |
| Evaluation       | Entscheidungen treffen<br>Auswerten von Daten<br>Konzepte vergleichen und zwischen ihnen unterscheiden<br>Den Wert von Theorien/Konzepten bewerten |

## Reguläre Ausdrücke

Die folgenden Informationen sind [Wei05] entnommen und sollen dazu dienen, die in Kapitel 5.3.2. genutzten regulären Ausdrücke für einen Leser, der sich noch nicht mit regulären Ausdrücken befasst hat, näher zu bringen. Die Nutzung regulärer Ausdrücke ist für die Tokenisierung und das Extrahieren vordefinierter Suchmuster in Texten sehr nützlich. Für eine tiefe Auseinandersetzung mit dem Thema bietet sich [Fri03] an.

„Reguläre Ausdrücke sind Suchmuster, die zur Analyse von Zeichenketten verwendet werden. Ein regulärer Ausdruck definiert eine Menge von Strings, auf die er ‚passt‘. Wenn  $re$  ein regulärer Ausdruck ist, so bezeichnet man die Menge der passenden Strings auch als Sprache des regulären Ausdrucks oder  $L(re)$ .“

**Tabelle 0-2 Die wichtigsten Sonderzeichen und -sequenzen in regulären Ausdrücken [Wei05]**

| Sonderzeichen | Bedeutung                                                                                                                                           |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| .             | Jedes Zeichen außer Zeilenwechsel ( $\backslash n$ )                                                                                                |
| ^             | Beginn eines Strings oder das erste Zeichen nach $\backslash n$ oder Bildung des Komplements einer Zeichenmenge                                     |
| \$            | Ende eines Strings und das letzte Zeichen vor $\backslash n$                                                                                        |
| *             | Beliebig häufiges (eventuell keinmaliges) Wiederholen des vorausgehenden regulären Ausdrucks                                                        |
| +             | Ein- oder mehrmaliges Wiederholen des vorausgehenden regulären Ausdrucks                                                                            |
| ?             | 1) kein- oder einmaliges Auftreten des vorhergehenden regulären Ausdrucks<br>2) vorausgehendes ‚+‘- oder ‚*‘-Zeichen wird als ‚gierig‘ qualifiziert |
| { $m$ }       | Genau $m$ -maliges Wiederholen des vorausgehenden regulären Ausdrucks                                                                               |
| \             | Maskieren eines Sonderzeichens                                                                                                                      |
| \A            | Beginn der Zeichenkette                                                                                                                             |
| \d            | Dezimalziffer, entspricht der Menge [0-9]                                                                                                           |
| \D            | Alle Zeichen außer Dezimalziffern                                                                                                                   |
| \s            | Ein Whitespace-Zeichen aus der Menge [ $\backslash t \backslash n \backslash r \backslash f \backslash v$ ]                                         |
| \S            | Alle Zeichen außer Whitespace-Zeichen                                                                                                               |
| \w            | Irgendein alphanumerisches Zeichen aus [a-zA-Z0-9]                                                                                                  |
| \W            | Irgendein nichtalphanumerisches Zeichen aus [^a-zA-Z0-9]                                                                                            |
| \z            | Ende der Zeichenkette                                                                                                                               |
| [ ]           | Definition einer Menge von Zeichen                                                                                                                  |
|               | Oder                                                                                                                                                |
| ( )           | Zeichengruppe                                                                                                                                       |
| (? )          | Setzen von Flags                                                                                                                                    |

Durch die Kombination der Sonderzeichen wird ein komplexerer regulärer Ausdruck erzeugt, z.B.:  $[a-zA-Z.]+|\d+[,|:]\d+|\w+$ .